DIGITAL CONTROL OF THE UTAH MIT DEXTEROUS HAND: INITIAL EVALUATION AND ANAYLSIS

THESIS

Lloyd W. Rainey III
Captain, USAF

AFIT GE ENG 90D-50

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/90D-50

DIGITAL CONTROL OF THE UTAH/MIT DEXTEROUS HAND: INITIAL EVALUATION AND ANAYLSIS

THESIS

Lloyd W. Rainey III
Captain, USAF

AFIT/GE/ENG/90D-50

Approved for public release; distribution unlimited

AFIT/GE/ENG/90D-50

Digital Control of the Utah/MIT Dexterous Hand: Initial Evaluation and Analysis

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Lloyd W. Rainey III, BSEE
Captain, USAF

December 13, 1990

Preface

Investigation into new areas of research requires the support and assistance of many individuals. My sincerest gratitude goes to Dr. Michael B. Leahy Jr. , who presented the problem and continued with encouragement and guidance throughout the thesis effort. His instruction and professional example are invaluable resources for the robotics research currently being performed at the Air Force Institute of Technology, and in the efforts of the robotic students. A special thanks to Dr. Steven K. Rogers for his assistance and instruction in the neural networks research and Dr. Curtis Spenny for his robotics instruction.

The backing of research efforts is as important as the work itself. Therefore, I express my thankfulness to the Armstrong Aerospace Medical Research Laboratory Robotic Telepresence Program without whose sponsorship such studies could not be conducted. My appreciation goes to Mr. Dan Zambon, of the Information Systems Laboratory, for his efforts required to keep the computers running and ever present assistance.

This research could not have been conducted without the reinforcement of fellow students. Thank you, Clayton Andersen and Paul Whalen, for your ideas, ears, helping hand, and especially your friendship. I thank my parents for their many sacrifices and constant encouragement. Especially, I am grateful to my wife, Dawn, and son, Weslie, without whose love, patience, and support this effort would not have been possible. In closing, I thank the Lord Jesus Christ who provided me with the individuals and whatever talents were necessary to conduct this research.

Lloyd W. Rainey III

Table of Contents

List of Figures

List of Tables

AFIT/GE/ENG/90D-50

Abstract

An experimental digital platform is developed as an environment on which to evaluate digital control strategies for dexterous manipulation with a pneumatically actuated tendon-driven manipulator. This environment is used to begin the study of advanced control methods that are suitable for providing the tracking accuracy required for grasping and dexterous manipulation with a pneumatically actuated tendon-driven end-effector. The digital platform consists of a PC/AT-386 and a single board MC68020 microcomputer in a VME chassis with shared RAM between the two processors to control the Utah/MIT Dexterous Hand (UMDH). The MC68020 controls the A/D and D/A access for the UMDH, while the ARCADE_HAND experimental control environment is hosted on the PC/AT-386 for user interface and control determinations. An Adaptive Model-Based Control (AMBC) algorithm is implemented and experimentally evaluated on the UMDH. Tracking performance is compared to the PD baseline controller of the ARCADE_HAND environment and evaluated for the requirements of human finger emulation. The evaluation includes compensation for unknown dynamics of the UMDH system, adaptability to unknown payloads, and multiple trajectory tracking capabilities. The superior tracking of the AMBC algorithm demonstrates the potential of the technique for emulation of human finger movement.

Digital Control of the Utah/MIT Dexterous Hand: Initial Evaluation and Analysis

I. Introduction

1.1 Motivation

One problem in gross motion robot control is how to provide dexterous hand motion. A solution to this problem is one requirement for realizing a manipulator capable of emulating human performance. The Air Force is interested in developing human performance capabilities in order to remove the human operator from the site of hazardous operations while still providing the cognitive capabilities required to perform the required tasks. A manipulator that is able to emulate human performance is one prerequisite for achieving Air Force Telepresence Program objectives. To meet the requirements of robotic telepresence, present research on gross motion control must be expanded into the area of dexterous hand motion.

1.2 Objective

A new major initiative of robotic research at the Air Force Institute of Technology (AFIT) is the development of technologies required for semi-autonomous dexterous manipulation. The goal of this thesis effort is two-fold. The first goal is to develop an experimental platform that provides the capability to develop and evaluate digital control strategies for dexterous manipulation with a tendon-driven manipulator. The second goal is to use the new experimental platform to begin the study of advanced control methods that are suitable for providing the tracking

accuracy required for grasping and dexterous manipulation with a tendon-driven end-effector.

1.3 Problem Statement

Research on gross motion control aspects of robotic telepresence has been ongoing at AFIT for the past four years. Those efforts have concentrated on the development and evaluation of control methods which may lead to a robotic arm emulating human arm performance. The latest direction of this research, toward the long range goal of robotic telepresence, is developing technologies required for semi-autonomous dexterous manipulation. To pursue this avenue of study a robotic system platform must be designed and implemented that provides grasping and manipulation capabilities. The Utah/MIT Dexterous Hand (UMDH), on loan from the Armstrong Aeromedical Medical Research Laboratory, is a tendon-driven robotic manipulator which was designed to provide such capabilities. The system capabilities of the UMDH must be enhanced in order to provide a suitable platform for digital control studies. Once the platform is established, a control system must be developed that provides dexterous motion to the UMDH.

The mechanism responsible for providing grasping and manipulation with a tendon-driven manipulator is the control system. Controlling a robotic manipulator is complicated by the coupled and nonlinear nature of robotic dynamics. The control of a tendon-driven manipulator is further complicated by the employment of tendons which are routed through the manipulator, and indirect drive systems which introduce additional nonlinearity and coupling. These complications make many present day control schemes inappropriate. Consequently, alternative control approaches must be developed and evaluated to provide the control capabilities needed to provide the accurate dexterous motion required for grasping and manipulation.

1.4 Scope

The foundation for this effort was laid by the previous studies of gross motion control for the PUMA 560 robot arm [18, 19, 20, 23]. The scope of this research was based on the tendon-driven UMDH (left-hand) and its control electronics. The UMDH was used as a case study because: the hand system was made available by the Armstrong Aeromedical Medical Research Laboratory, the dynamics are not well-known, and it is a good case study for human finger movement emulation. At the start of this research, the capabilities at AFIT provided only for analog control of the UMDH. In order to facilitate the development and evaluation of alternative control approaches, a digital capability must be provided. The initial task was the development of the digital control computer system for the UMDH.

Once the digital control computer system was developed, the focus changed to determining a controller for the UMDH that is suitable for human finger movement emulation. Experimental studies of control systems for robotic devices with unknown system models are scarce and simulation studies are inadequate. Consequently, all design and analysis work was based on experimental data. In order to reduce the complexity of the design and analysis, only one finger of the UMDH was used in the experimental evaluations. The procedures to achieve gross motion control of one finger are directly transferable to the other fingers of the hand.

One proven form of robotic control is model-based control. This control method provides excellent trajectory tracking performance when an accurate system model of the manipulator is available [20, 2, 14, 15]. However, an accurate system model is not always attainable, as is the case for the UMDH. Adaptive model-based control approaches have been proposed as a means to provide improved control accuracy for robotic systems without significant a priori knowledge of system system dynamics [36, 39]. The unknown robotic dynamics are trained by adaptive mechanisms to provide a best-fit model for the system. These control systems offer promise for controlling highly nonlinear and coupled manipulators and have proven successful

in the control of the PUMA 560 industrial manipulator at AFIT [21]. The ability of Adaptive Model-Based Control (AMBC) to function without a priori knowledge and to adapt to dynamic uncertainties make it a promising choice for the UMDH.

Recent application of artificial neural networks to system control, especially in the area of robotics, offers an alternative solution for control of nonlinear and highly coupled robotic systems. Capt. Mark Johnson developed the Adaptive Model-Based Neural Network Control (AMBNNC) technique which uses artificial neural networks to provide a payload invariant controller [13]. W. Thomas Miller III at the University of New Hampshire developed a robotic controller using the principles of the Cerebellar Model Articulation Controller presented by J. S. Albus [1] in parallel with a fixed-gain controller and implemented the controller on an industrial manipulator [28, 29]. Another system controller using the Adaptive Linear Threshold Element (ADALINE) artificial neural network proposed by Bernard Widrow [44] was developed to both model and control a truck-backer-upper [31]. This was done in simulation and the principles hold promise for the robotic manipulator control system required to control a tendon-driven manipulator.

1.5 Method of Approach

Initially, the digital control computer system for the UMDH was developed and evaluated. The analog controller of the UMDH was bypassed and a new digital computer control system environment implemented. Once development of the digital computer system was completed, a revised version of AFIT's ARCADE computer environment [18], ARCADE_HAND, was rehosted onto the PC/AT-386 for the left-handed version of the UMDH. ARCADE_HAND provides the ability to select test conditions such as the fingers to be controlled, the control algorithm used, and what error data to generate and store. The capability to control all four fingers simultaneously was provided, although present hardware limitations prevent this from becoming a reality until a future date.

With the ARCADE_HAND support available, simple finger Position-plus-Derivative (PD) loops were developed and experimentally evaluated as the baseline controller for the digital environment. The PD algorithm has been explored on the PUMA 560 and adjustments were made to adapt the control capabilities to the robotic hand. From the baseline PD environment an alternative control method was developed and evaluated. Recent in-house studies on robotic control methods have indicated promising results from AMBC [21]. Consequently, the AMBC algorithm used on the PUMA 560 was adapted for implementation on the UMDH digital control system. The AMBC algorithm is developed and evaluated for its suitability for human finger emulation. Efforts were initiated to develop an Artificial Neural Network Controller for implementation on the UMDH, but time constraints did not permit experimental evaluation.

1.6 Materials and Equipment

The following items have been acquired on loan from AAMRL/BBA:

- Utah/M.I.T. Dexterous Hand (left-hand)
- SARCOS control electronics for the hand
- Ironics IV-3273 System Controller
- Ironics IV-3201 VME-bus Multiprocessing Engine
- Data Translation A/D and D/A Converters
- BIT3 IBM PC/AT VME Adaptor

An IBM-386 PC is used to interface with the above equipment.

1.7 Contributions

Completion of the objectives for this thesis mark a significant contribution to AFIT's gross motion control studies. The digital control environment enables future

dexterity and manipulation studies on the UMDH at AFIT. The AMBC control algorithm was applied to the tendon-driven UMDH and experimentally evaluated for applicability to dexterous motion control and and human finger movement emulation. This thesis effort indicated that Adaptive Model-Based Control schemes are applicable for emulation of human finger movements on a tendon-driven manipulator and that similar types of algorithms might be used for future studies.

II. Literature Review

2.1 Introduction

Research on gross motion control aspects of robotic telepresence has been ongoing at AFIT for the past four years under the broader aegis of the gross motion control project. A new major initiative of AFIT robotic research is developing technologies required for semi-autonomous dexterous manipulation. These efforts are the driving force behind the dexterous motion control project on the Utah/MIT Dexterous Hand (UMDH). This research develops a digital control environment for the evaluation of alternative approaches to provide the dexterous motion required for grasping and manipulation. Also, the first application and evaluation of a modern control approach on a UMDH is conducted. The key components of this research are the manipulator used for implementation of these control approaches and the control methods employed to provide the required motion. The manipulator used for this research is the UMDH. This unique platform, which was developed for research on dexterity and manipulation, is discussed in detail in the next section. Following this discussion is a review of the present status of robotic manipulator control research and the applicability of those control methods to the UMDH.

2.2 Utah/MIT Dexterous Hand

The platform for this study is the UMDH [12, 11, 4, 10]. This is a tendon-operated multiple-degree-of-freedom robotic hand that is designed to provide an understanding of important issues related to machine-based artificial dexterity. Figure 2.1 shows a picture of the UMDH. The dexterous hand is approximately the same size and geometry as the human hand. It includes three, rather than four, four-degree-of-freedom fingers and one four-degree-of-freedom thumb. Other components of the system are the palm, actuators, and sensors.

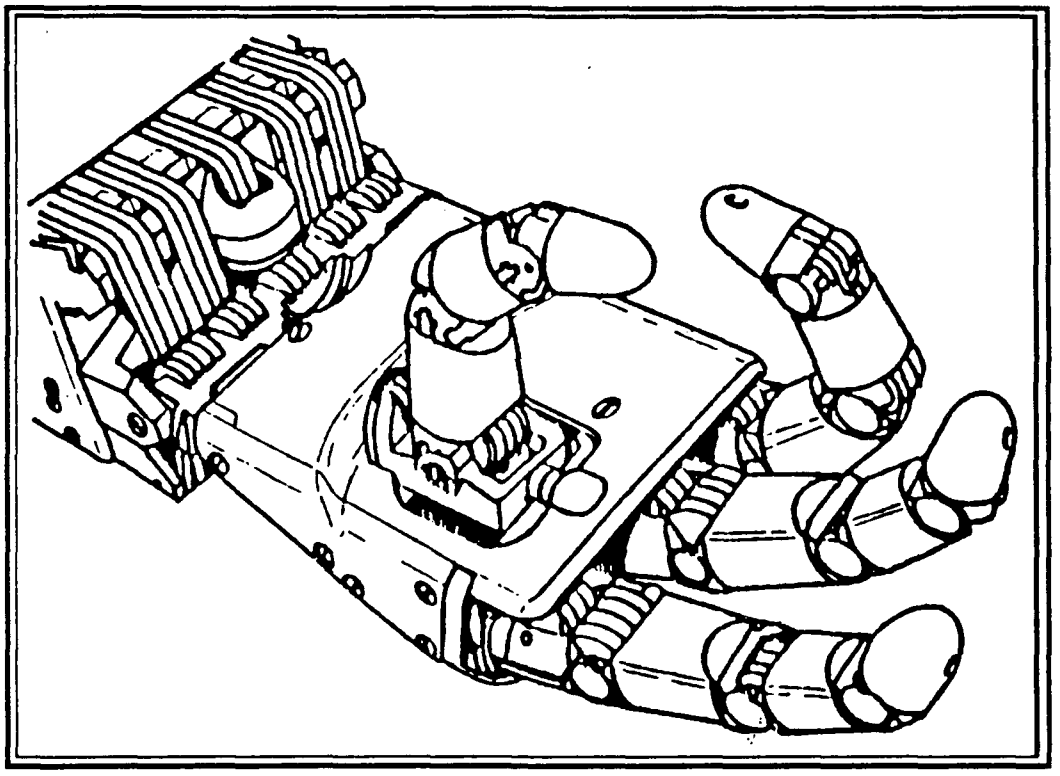

Figure 2.1. The Utah/MIT Dexterous Hand

The three fingers and thumb act together to provide manipulation capability similar to the human hand. The three fingers are arranged in a planar sequence to allow formation of an adaptable surface against which the thumb can act. The four degrees of freedom are provided by four joints in each finger. These joints consist of three parallel axis joints to provide curling action and a fourth proximal joint near the palm, which is perpendicular to the other axes, provides side-to-side motion of the finger. The three curling joints provide 0-95 degrees of excursion and the proximal joint allows for lateral excursions of plus or minus 25 degrees. The thumb also contains four active degrees of freedom and is configured to provide approximately human-like motions during lateral and palmer grasps. The thumb's proximal joint is capable of plus or minus 45 degree motions.

The palm is the structural mounting base for the fingers, thumb, and wrist. It also provides a transition region for the tendons from the fingers and thumb that must pass through the wrist to the actuation systems located on the forearm. For multiple finger movements the palm provides a convenient reference frame. Unlike the human hand, the manipulator's palm has no degrees of freedom. It is completely stationary.

The tendons connect the joints to the actuators, which provide the power for the system, much as our tendons connect from bone to muscle. The tendons are composed of a special Kevlar and Dacron composite. The axial Kevlar fibers support tension loads, and the Dacron mat, which is interwoven with the Kevlar, provides abrasion protection for internal structures. The tendon system for the four joints of each finger and thumb consists of eight tendons acting together as antagonist pairs, one pair per joint. The tendons are routed over pulleys and run from an actuator to the link to which it is clamped.

The most important factor that determines mechanical performance of the dexterous hand is the actuator performance. The dexterous hand requires 32 separate actuators which produce the tendon tensions. A pneumatic system is used for the

actuators. The cylinders are ground-glass tubes with a 1.6 cm internal diameter that have a graphite piston to provide a stroke of 3.2 cm. A small-diameter tension rod is connected through a low-friction seal to provide tendon connection. The cylinders are configured in a close pack 4 x 4 hexagonal array and stacked in two offset layers to provide the 32 tendon actuators in a single package.

The sensors are the component that gives the system a sense of "intelligence" and provides the information required for accurate control. Each joint contains a sensor to measure angular deflection. These sensors are magnetically sensitive Hall effect devices located in the proximal links. Also, each tendon has a tension sensing system located in the wrist which is a semiconductor strain gauge that detects beam strain due to load on the pulley from the tendon.

The UMDH is a useful test bed for fine motion control and dexterity studies. The high degree of coupling and significant nonlinearities inherent in the structure make it unsuitable for many of the more popular control systems in use today. Consequently, it is necessary to develop and evaluate some alternative approaches.

2.3 Control Methods

Throughout the past decade, many manipulator control schemes have been studied for applications on robotic devices. However, for various reasons many have serious drawbacks in the control of a tendon-driven manipulator. These control schemes can be broken up into three categories: Classical, Adaptive, and Artificial Neural Network Controllers. Each of these categories has its own control implementations. Some of these implementations are discussed and their applicability to gross motion control of the UMDH is evaluated.

2.3.1 Classical Controllers

2.3.1.1 Fixed-gain Feedback and Feedforward Control The simplest and most common form of robot control is fixed-gain feedback control [15, 2]. This is

often implemented in the form of independent joint Position-plus-Derivative (PD) or Position-plus-Integral-plus-Derivative (PID) control of the form

$$\tau = K_v(\dot{q}_d - \dot{q}) + K_p(q_d - q) + K_I \int (q_d - q).$$

This control has good endpoint accuracy but the tracking performance is poor, especially at high speeds.

An augmentation to this control method is the model-based technique of feedforward dynamics compensation. This control method, often called Simple Model-Based Control (SMBC), includes a set of nominal torques based on the robotic system dynamics. The feedforward torques are calculated based on the desired state variables and are fed forward to be added to the linear feedback terms to linearize the system about the desired operating points. The feedforward terms can be computed off-line since they are based on the desired trajectory. Although this augmentation can significantly improve performance, it is limited by the accuracy and completeness of the system model. Also, the discrepancy between the desired and actual trajectory is not taken into account when calculating the feedforward terms [20, 15, 2].

2.3.1.2 Computed Torque Control Computed torque controllers compute the dynamics on-line and make use of the sampled joint position, velocity, and/or acceleration data in the calculations [24, 14, 15, 2]. This scheme utilizes nonlinear feedback to decouple the manipulator. Computed torque controllers use a dynamic model of the robot to calculate joint drive torques based on the measured state variables for the specified trajectory. Although this method provides excellent results when the complete dynamics of a manipulator are known, this is only true for a restricted set of manipulators. Computed torque controllers usually handle 3 to 5 actuators and have proven successful on industrial manipulators such as the PUMA 560 [18]. As previously mentioned the UMDH employs 8 control actuators per finger, thus significantly complicating the dynamic model and trajectories. Also,

development of a dynamic model for the hand platform would be nearly impossible due to the "stretchy tendons", increased joint friction, and other nonlinearities such as the compliance of the pneumatic actuators [4, 12]. Introduction of uncertainties in payload, obstacle interference, and multiple tracking trajectories severely degrade performance of the computed torque control scheme making it inapplicable to this platform.

2.3.2 Adaptive Control

2.3.2.1 Model-Reference Adaptive Control Model-Reference Adaptive Control (MRAC) techniques have been introduced as a way to account for uncertainties in the system model [34, 39, 36]. These methods no longer rely on previously determined manipulator dynamics but instead use a general model and attempt to modify the torque values based on position, velocity, and acceleration errors. Seraji presented the development of a decentralized Lyapunov-based MRAC (LB-MRAC) and implemented this controller on a PUMA 560 to support his claims of improved performance [37, 36]. The proposed control scheme does not use the complex robotic dynamic model. Instead, each joint is controlled simply by a PID feedback controller and a position-velocity-acceleration feedforward controller, both with adjustable gains. At the Air Force Institute of Technology (AFIT) experimental evaluation was done of the decentralized LB-MRAC algorithm proposed by Seraji. The algorithm's performance was inferior to a model-based controller with fixed PD gains. The algorithm was also unable to compensate for payload variations, a supposed advantage of LB-MRAC. The conclusion was reached that centralization was required [23].

2.3.2.2 Adaptive Model-Based Control Slotine and Li have proposed an adaptive robot control algorithm which consists of PD feedback and full dynamics feedforward compensation, with the unknown manipulator and payload parame-

ters being estimated online [39]. The approach to Adaptive Model-Based Control (AMBC) proposed by Slotine and Li uses parameter adaptation based on Lyapunov theory to compensate for unknown robotic system dynamics. This algorithm was successfully demonstrated on the MIT WAM robot [32]. Recent simulation studies suggest that this algorithm is unstable in the presence of velocity measurement noise [21].

Sadegh and Horowitz proposed a variation on the AMBC approach which eliminates the velocity measurement problem [33]. The regressor matrix in their "Desired Compensation Adaptation Law" is strictly a function of the desired trajectory values. This revised AMBC algorithm has been successfully evaluated at AFIT on the PUMA 560 and a wide range of implementation issues investigated. Results indicate a significant performance improvement and show that a priori knowledge of dynamics is not required if learning is permitted [21]. As a result of the success of the AMBC on an industrial manipulator without a priori knowledge of system dynamic parameters, AMBC is a promising candidate for application on the UMDH. An excellent tutorial on various forms of AMBC is in the Spong and Ortega, Adaptive Motion Control of Rigid Robots: A Tutorial [34].

2.3.3 Artificial Neural Networks Artificial Neural Networks (ANN) models are composed of many computational elements, often nonlinear in nature, operating in parallel and arranged in patterns reflecting those of biological neural networks. The computational elements are connected via weights that may be adapted during use, that is training, to improve performance [25, 45]. The basic premise in applying artificial neural networks to robot control is to use the network to learn the characteristics of the robot system, rather than specify explicit robot system models. Although there is widespread interest in this problem within the neural network and robotic communities, few theories have been validated by actual robot control experiments. This lack of application is due to the computational speed and stability

problems that result when employing neural networks of sufficient complexity for realistic robot control problems [29].

2.3.3.1 Cerebellar Model Articulation Controller Many control schemes have been proposed for neural networks and one in particular has been applied to a complete robotic system. This control scheme was first proposed by J. S. Albus as the Cerebellar Model Articulation Controller (CMAC) [1]. CMAC is an adaptive control system which applies control functions for many degrees of freedom operating simultaneously by referring to a table rather than mathematical solution of simultaneous equations. Input commands and feedback variables are combined into an input vector which is used to address a memory where the appropriate output variables are stored [17].

The control scheme presented by Albus was adapted by Miller et al. into a control scheme that is quite different from that proposed by Albus but employs the CMAC principles [28, 29, 16]. Miller's controller is similar to the computed torque controllers, however the robot dynamic model is replaced by a neural network model. A training scheme adjusts the CMAC network on-line based on observations of the input and output relationships to form an approximate dynamic model of the robot in the regions of operation. The CMAC network is used to predict the control torques required to follow a desired trajectory, and these torques are used as feedforward terms in parallel to a fixed-gain linear feedback controller. These studies present successful results of real-time experiments which involved learning the dynamics of a five-axis industrial robot (General Electric P-5), during high-speed movements simulating industrial tasks. Of note here is that the study was for movements simulating industrial tasks. There is a low degree of coupling in the five-axis industrial robot that was used and no payload variation was attempted. The payload variation had been previously simulated. However, one particularly positive aspect of the CMAC control scheme is that it is structured in such a way as to be able to incorporate less conventional sensor inputs such as touch sensors and vision

systems [27]. It is necessary to determine if similar applications of neural networks can succeed on more general applications and for a less restricted range of operation.

2.3.3.2 Adaptive Model-Based Neural Network Control Another application of artificial neural networks was the work of Capt. Mark Johnson [13]. He applied neural networks to a PUMA 560 manipulator to provide payload invariant control. The neural network employed was able to identify the unknown payload on the manipulator by the trajectory tracking error and correct feedforward control torques by adjusting the system model. Thus, the neural network was able to adapt the control scheme to the uncertainty of the manipulator payload. This technique is not appropriate for the UMDH because it requires a priori knowledge of the manipulator dynamics.

2.4 Summary

The requirement to provide fine dexterous manipulative control to the Utah/MIT Dexterous Hand is no small task. Most control schemes developed to date are restricted by trajectory, model dynamics, or other uncertainties. Adaptive Model-Based Control (AMBC) provides outstanding tracking performance and other desirable manipulator capabilities on a PUMA 560 with limited a priori knowledge. These traits could make AMBC a suitable alternative for the UMDH. Artificial neural networks, in particularly CMAC, also offer the potential of solving some of the control problems that are encountered on today's manipulators, especially one as highly coupled and nonlinear as the UMDH.

III. Digital Controller Development

3.1 Problem Analysis

Gross motion control efforts at the Air Force Institute of Technology (AFIT) are moving into the realm of dexterous manipulation. The platform for these studies is the Utah/MIT Dexterous Hand (UMDH). The control system capabilities for the robotic hand system have been limited to the internal analog control system provided by the manufacturer of the robotic hand system, SARCOS Inc. [9]. In order to expand the avenues of future dexterous manipulation studies on this platform, a new digital control system environment must be developed for the UMDH. The first objective was to create the experimental environment for the robotic hand system. The resultant system must support the experimental evaluation of modern control methods. Once the environment was established, the baseline Position-plus-Derivative digital controller was developed and evaluated.

3.2 Experimental Environment

The left-hand version of a UMDH, operating under a revised version of the ARCADE environment [18], is the target platform of the control system development for future dexterous manipulation studies. The UMDH electronics include an internal analog control system which can be bypassed for external implementation of a digital control system. The internal analog controller does not provide the tracking performance or other dexterity requirements needed for emulation of human finger movement. For this reason the analog control system is completely bypassed and an external digital control system must be developed and implemented.

The adapted version of the ARCADE environment, ARCADE_HAND, is resident on an IBM PC/AT-386 which uses a serial interface and shares dual-port memory RAM with an IV3201 real-time processing engine by IRONICS Inc. which

is based on a 16MHz MC68020 processor [8] and is resident on the UMDH VME-bus. Figure 3.1 illustrates the control architecture of the robotic hand system and Figure 3.2 provides a chart of the software operation. The pertinent components of the newly developed robotic hand control system can best be described by stepping through the operation sequence. Operation is initiated when the user downloads a read/write program to the MC68020 through the PC/AT communication port via the serial interface. Execution of this program begins when a command sequence is sent through the same port. An IBM PC/AT-VME adaptor provides a RAM piggy back card as a dual port memory which both the PC/AT and the VMEbus systems can access. The 128K byte RAM is mapped into unused address space on the two buses.

The two processors are independent of one another, however, semaphores synchronize the exchange of data through the RAM. The data in the RAM is organized in a C structure called the BIT-3 Window. This structure is identical in the software package of each computer and is assigned a designated address in RAM for each computer system. The MC68020 processor commands a set of A/D converters whose inputs are from the UMDH electronics. The processor reads the data for each digit from the A/D outputs and writes the velocity, position, and flexion and extension tension data to the shared RAM. The Intel 80386 processor reads the stored state variables of the system and computes the desired torques based on the data from the previous sample period, $T_s(k-1)$. The calculated torques are then converted to tendon tensions. The tensions are written from the PC/AT to the shared RAM, read by the MC68020, and then written by the MC68020 to the D/A converters. The converter output provides tension commands to the manipulator. The independence of the processors allows the 80386 to calculate the tendon tensions while the MC68020 is reading the next system state, and the 80386 to read the state variables while the MC68020 is commanding the tendon tensions.

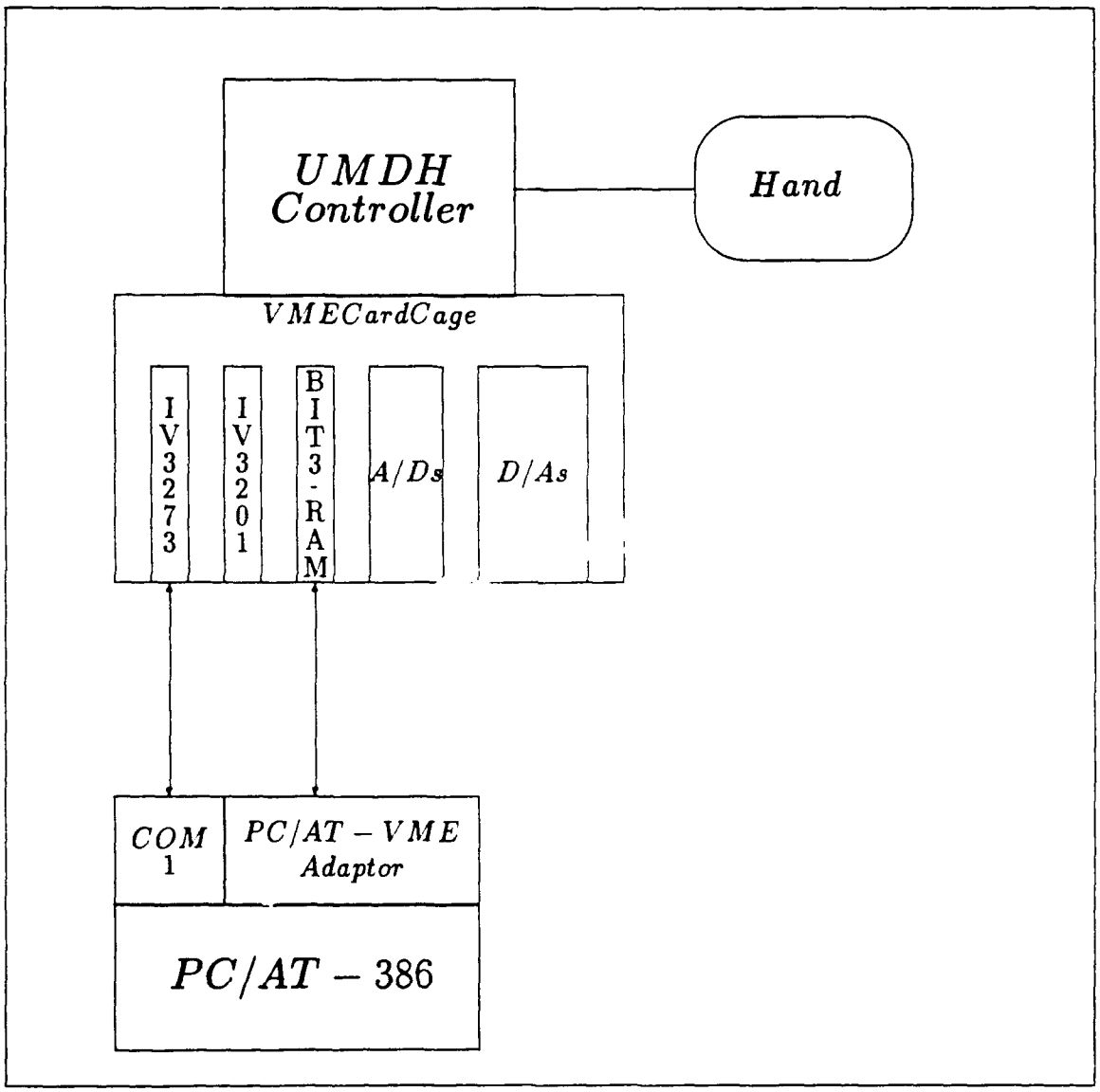

Figure 3.1. Control System Hardware

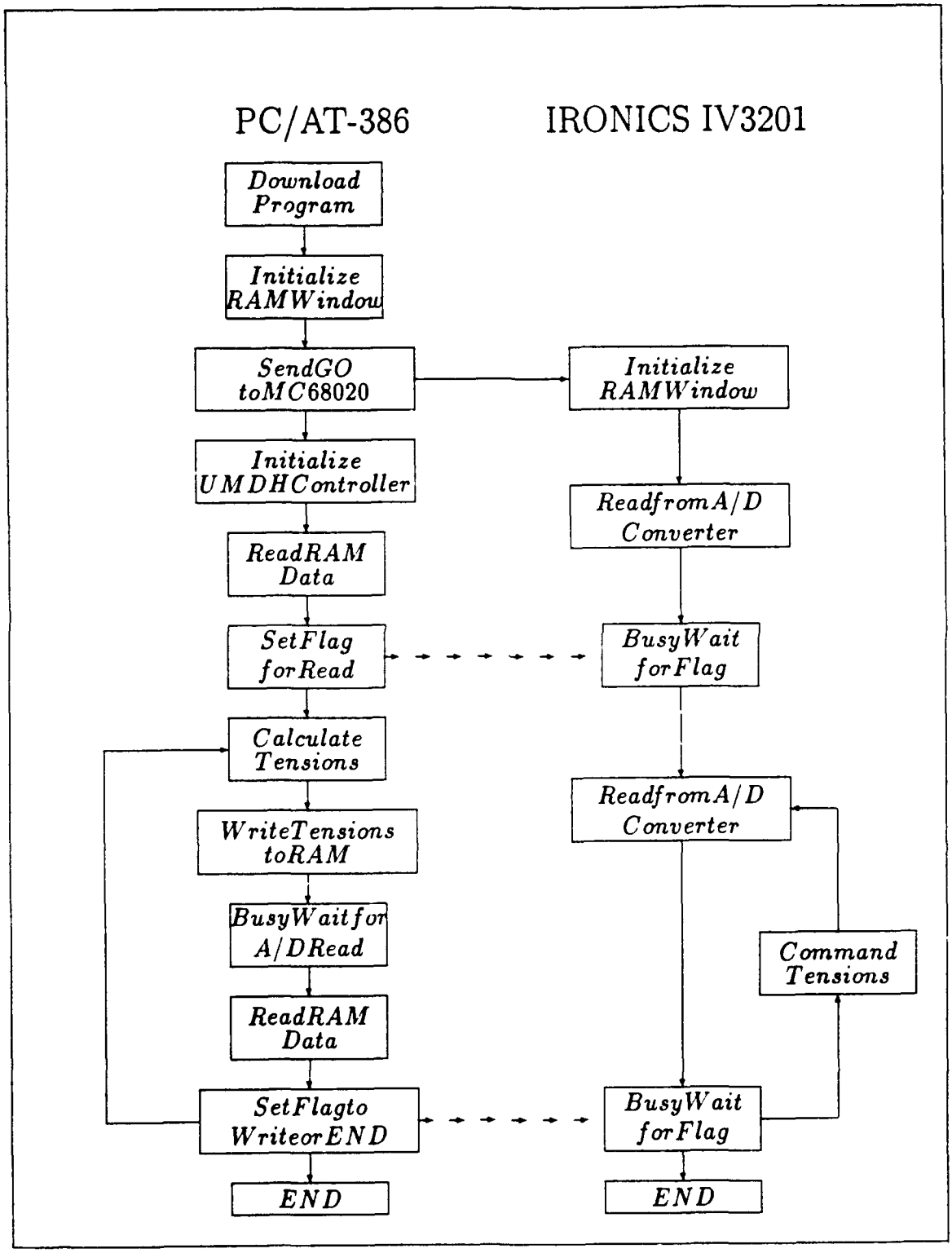

Figure 3.2. Control System Software

The ARCADE environment, written in FORTRAN [18], was converted to C for the ARCADE-HAND environment resident on the PC/AT and was compiled using Turbo C. The read/write program resident on the VMEbus is also written in C and was compiled for the MC68020 with an Aztex-C compiler [41]. Detailed program listings are in the internal report generated by this thesis [7].

Theoretically all fingers of the UMDH have similar dynamics. With this in mind, the testing platform for the robotic manipulator was reduced to Joints 2 and 3 on Finger 3 of the UMDH for analysis of the digital controller and PD control algorithm. These two joints were chosen over Joints 1 and 2 because of the difficulty involved in maintaining Joint 3 in a fixed position throughout a trajectory involving Joints 1 and 2. All three joints were digitally controlled but control algorithms were only applied to Joints 2 and 3. Joint 1 was fixed stationary by commanding appropriate constant tensions. The chosen test configuration for Joints 2 and 3 thus resembles the two degree-of-freedom elbow manipulators frequently used in control simulation studies. This choice was made to simplify the analysis of the control system and maintain a sample period adequate for control purposes. The control system is capable of performing at a sample period (T_s) of 2.5 milliseconds for an entire finger, thus providing 400 Hz operation. The later addition of the Adaptive Model-Based Control (AMBC) algorithm to the baseline PD controller increased the required computation time and resulted in a change in the sample period, T_s. The sample period was increased to 3.0 ms, thus operating at 333 Hz. This speed still proved capable of adequately sustaining manipulator performance. The sample period is quickly increased further if additional fingers are included in the algorithm. The primarily limitation of the sample period is the access times of the A/D and D/A converters and the set-up time required to execute the A/D and D/A access. Table 3.1 lists the present capabilities of the Read and Write access times to the A/D and D/A converters. Reading involves the collection of position,

velocity, and extension and flexion tension data from the A/D converter. Writing involves commanding extension and flexion tensions to the D/A converter.

Table 3.1. Read and Write Access Times

READ (ms)	WRITE (ms)
Hand 4.3	Two Fingers 0.9
Finger 1.7	One Finger 0.6
Joint 1.3	Joint 0.5

3.3 Baseline Digital Controller Development

The output tension vector (τ) of the digital controller is of the form

$$T = T_M \tau + T_{co}. \tag{3.1}$$

The output tension vector is composed of the transformation matrix (T_M) of the robotic finger, the total torque vector (τ) which consists of feedforward (τ_{ff}) and/or feedback (τ_{fb}) torques, and the cocontraction torque vector (τ_{co}). The sign of the output torque vector components indicate whether the magnitude is to be applied as flexion or extension torques. The transformation matrix accounts for the coupling introduced by the overlapping tendon routing through the robotic finger. The matrix is of the form

$$T_M = \begin{bmatrix} 1/R_1 & -1/R_2 & 0 \\ 0 & 1/R_2 & -1/R_3 \\ 0 & 0 & 1/R_3 \end{bmatrix} \tag{3.2}$$

where R_1 and R_2 are the pulley radii of Joints 1 and 2, respectively. The cocontraction torque vector (T_{co}) is introduced to keep a minimal tension on the tendons at all times. This prevents slack in the tendons, which otherwise might result in the tendons falling off of their pulleys. Impedance is controlled by (T_{co}). The cocontraction torque vector also keeps the joints from drifting in either the flexion or extension

direction due to a lack of force opposing the motion. The cocontraction values selected for the digital controller were selected to optimize the tracking performance and are not necessarily optimum for controlling drift. The actual values for T_{co} are 10.0 Newtons for Joint 2 and 15.0 Newtons for Joint 3.

The main component of the output tension vector (T) is the total torque vector (τ). The feedforward torque vector (τ_{ff}) component of τ is initially set to zero for the digital controller environment. For the feedback torque vector (τ_{fb}) component of τ a Proportional-plus-Derivative algorithm is implemented. Proportional-plus-Derivative is the simplest control algorithm which can provide adequate tracking response and asymptotic stability for a robotic manipulator [3]. The general form of the Proportional-plus-Derivative feedback algorithm is given by

$$\tau_{fb}(t) = K_D\dot{e}(t) + K_P e(t). \tag{3.3}$$

The position error vector of a single joint is the calculated difference between the desired and actual angular position as shown by Equation 3.4. The velocity error vector is given by Equation 3.5 as the difference between the desired velocity and the time differenced calculation from the position of the previous cycle. The sample period is T_s.

$$e(t) = q_d(t) - q(t). \tag{3.4}$$

$$\dot{e}(t) = \dot{q}_d(t) - [q(t) - q(t-1)]/T_s \tag{3.5}$$

The position and velocity gains are represented by the constant diagonal matrices, K_P and K_D respectively.

The delay inherent in digital implementation is handled by using error information from the previous sample time in the current cycle output torque calculations. The position error, velocity error, and feedback torque vectors then become

$$e(k-1) = q_d(k-1) - q(k-1), \tag{3.6}$$

$$\dot{e}(k-1) = \dot{q}_d(k-1) - [q(k-1) - q(k-2)]/T_s, \tag{3.7}$$

and

$$\tau(k) = \tau_{fb}(k) = K_D\dot{e}(k-1) + K_P e(k-1). \tag{3.8}$$

The position error is tested by commanding each of the last two links of the robotic finger through 60 degree trajectories in 1.2 seconds. The original trajectories are executed in the horizontal plain to exclude gravity effects; that is the hand is placed on its side. Joint 0 is held motionless by the internal analog controller of the UMDH , while Joint 1 is held as firmly as possible in the extended position by the digital controller. The digital controller must be used to maintain Joint 1 in a stationary position because the analog controller is unable to hold Joint 1 in place while Joints 2 and 3 run through their trajectories. The initial position of the robotic finger is full extension and the final position is Joint 1 extended, and Joints 2 and 3 at 60 degrees. The PD controller moves the robotic finger to its initial position, at which time an algorithm can be selected to track the desired trajectory. A minimum jerk trajectory generator was used to calculate the test trajectory [18]. Figure 3.3 shows the position, velocity, and acceleration profiles. The velocity and acceleration achieved by the UMDH with this trajectory are significantly higher than those of larger manipulator platforms such as the PUMA 560 [24].

3.4 Baseline Digital Controller Evaluation

The Proportional-plus-Derivative feedback loop gains of Equation 3.8 used during testing are listed in Table 3.2. These values were determined by numerous

Table 3.2. PD Feedback Gains

Link i	Position(K_P)	Velocity(K_D)
2	7.5	4.5
3	0.045	0.030

experimental runs. Stiffer gains were initially attempted for the PD controller but

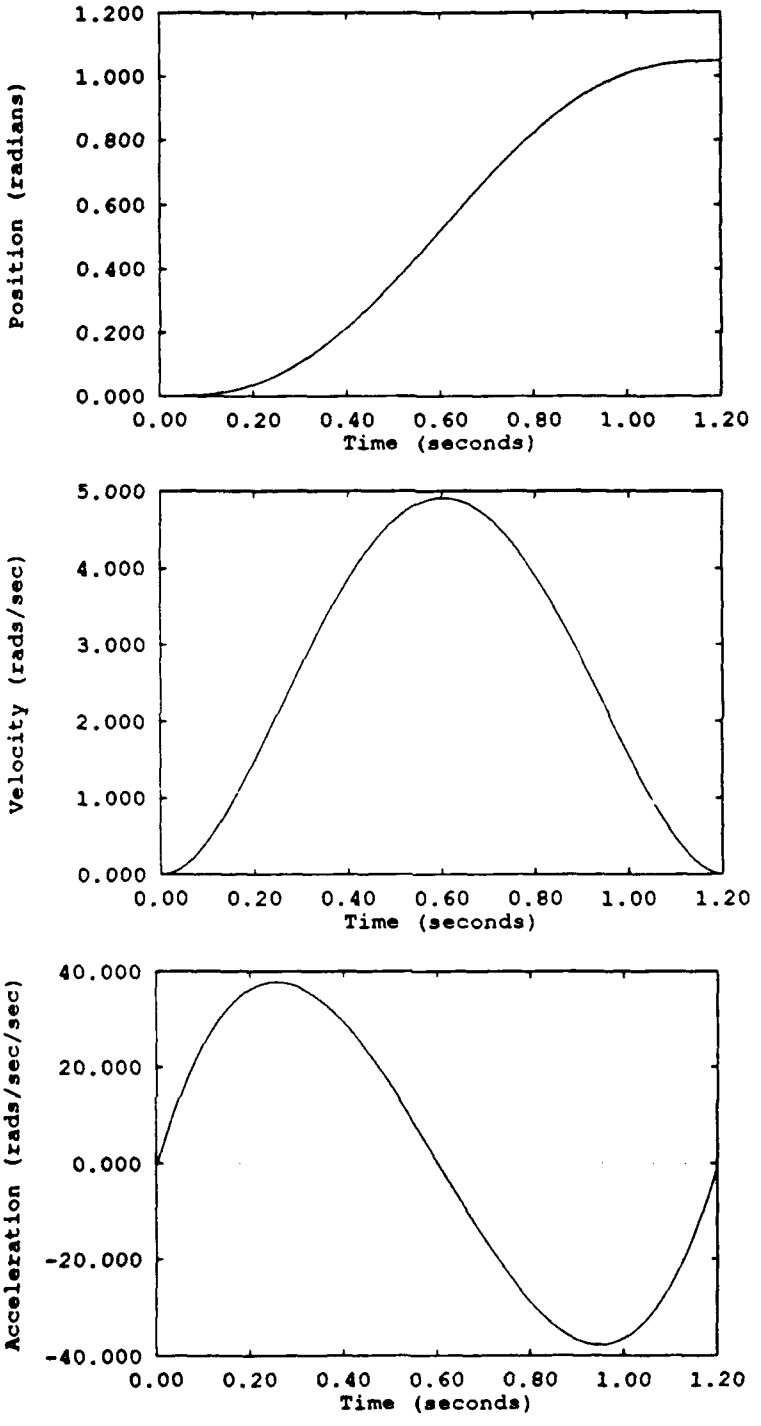

Figure 3.3. Original Trajectory Profile

the initial tracking error due to stiction amplified to produce a torque that was so severe the finger went to full flexion before the controller could respond to slow down the exaggerated response. The PD gains used produced the best case tracking error shown in Figure 3.4. The oscillatory tracking behavior of the joints, especially for Joint 3, is consistent throughout this thesis. This behavior is attributed to over-driving the output torques of the joints. The Recommendation Section of Chapter 5 addresses some possible solutions for reducing the oscillations.

The PD controller works for point to point control but is inadequate for the high speed tracking of which the UMDH finger is capable. This makes the PD controller unsuitable for dexterous manipulation on the UMDH. Therefore, the PD digital controller of the ARCADE_HAND environment is used to provide the foundation of the UMDH digital control system while new control methods are investigated for future studies of dexterous manipulation.

3.5 Summary

A digital controller evaluation environment was developed for the Utah/MIT Dexterous Hand. A PC/AT-386, an IRONICS MC68020 based single board computer, and A/D and D/A converter cards in a VMEbus are the major components of the digital control system. A dual port RAM is used to transfer data and commands between the two processors. ARCADE has been adapted to this platform as ARCADE_HAND. The baseline control algorithm in the ARCADE_HAND environment is a simple Position-plus-Derivative (PD) controller. The control environment is resident on the PC/AT. The ARCADE_HAND control environment provides the platform from which future studies in dexterous manipulation can build.

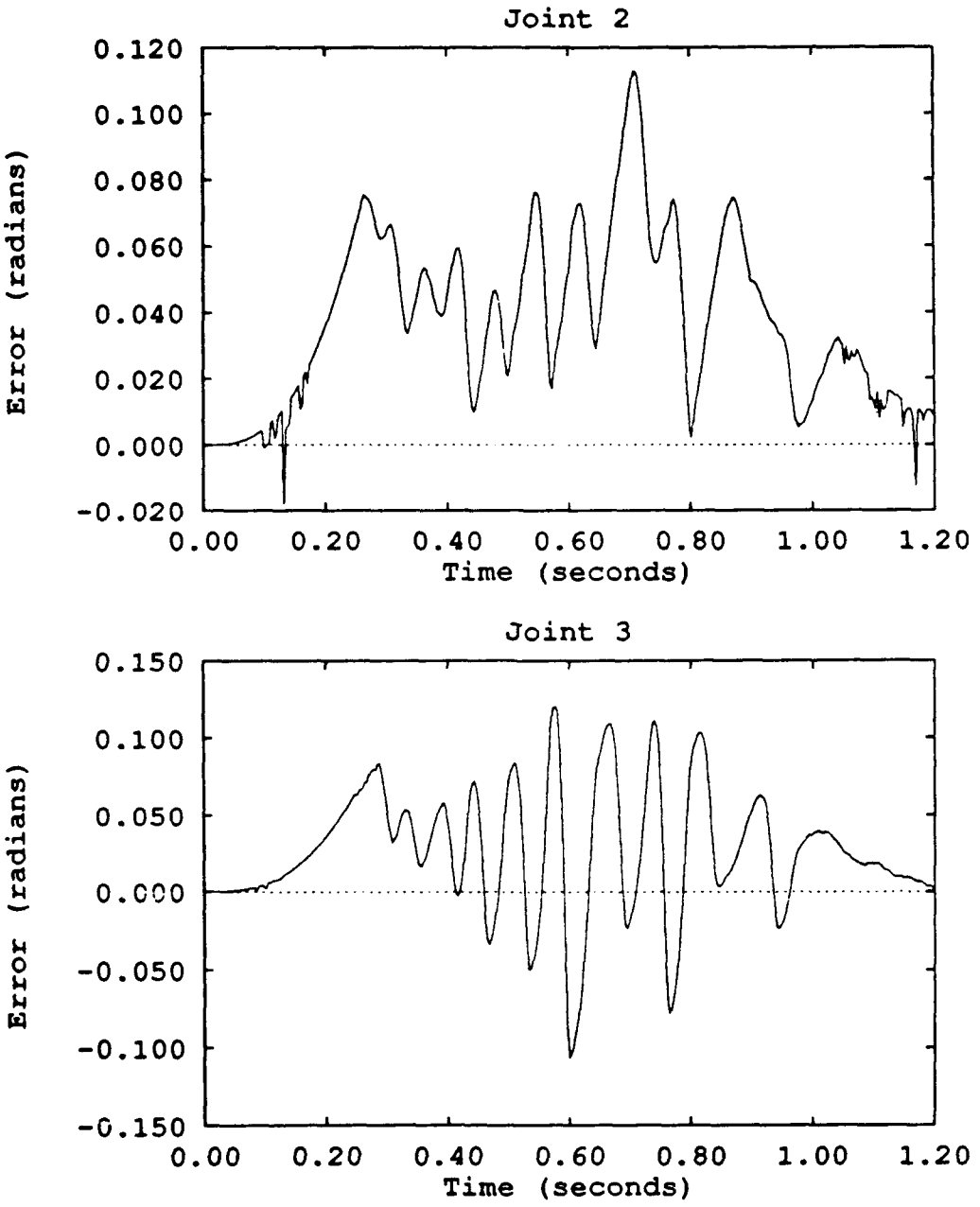

Figure 3.4. PD Feedback Tracking Error

IV. AMBC Development and Evaluation

4.1 Introduction

Central to providing for the grasping and manipulation of a tendon-driven dexterous end-effector is the development of an adequate control system. The initial task in that development is to determine the requirements of that control system and then what type of control algorithm best fulfills those requirements. The following is a list of baseline requirements that must be addressed by the control algorithm of choice [3, 22, 20].

- High speed tracking,
- Accurate end-point performance,
- Multiple trajectory tracking, and
- Payload invariance.

In order to meet these requirements, the control system must address several issues inherent in the construction of a tendon-driven robot. The controller must compensate for the disturbances produced by the coupled and nonlinear nature of robotic link dynamics. Included in link dynamics are Coriolis and centrifugal forces, inertial forces, and gravitational forces. The indirect tendon-drive system used in the UMDH also produces significant disturbances due to viscous and coulomb friction, tendon elasticity, pneumatic actuation effects, and tendon routing, further complicating the overall robotic system dynamics.

The Single Model-Based Controller (SMBC) has been proposed to address many of the issues listed above. This control algorithm has been implemented to solve the compensation problems for direct-drive [2, 14, 15] and industrial manipulators with high torque amplification drive systems [24, 19, 21]. However, the SMBC control system requires a priori knowledge of the manipulator dynamics, payload, and

trajectory to function effectively [24]. The dynamics of the UMDH do not lend themselves to such availability. The robotic hand is also on a much different scale than the direct-drive and industrial manipulators on which SMBC was previously evaluated. The small scale moving parts of the robotic hand do not induce the dominant inertial and gravitational effects of larger robotic manipulators. Instead, the pneumatic indirect-drive system and the tendon peculiarities become the dominant dynamics. The compression and inertial effects of the pneumatic actuators, along with the elasticity and coupling of the tendons have not been, and probably cannot be, accurately modeled. Consequently, despite SMBC's success on industrial manipulators, it is unable to compensate for the dynamic effects on the UMDH, and thus does not meet the requirements as a controller of this robotic manipulator. Therefore, a different control system algorithm must be considered.

Adaptive Model-Based Control (AMBC) has been proposed as a control system which more completely addresses the issues in question. As with SMBC, this control system has also been used to solve the control problems for direct-drive [6, 38, 30] and industrial manipulators [37, 42, 19, 43, 26, 21]. However, AMBC is able to compensate for unknown nonlinear dynamic effects in robotic systems [39, 34]. Application of AMBC methods to industrial manipulators such as the PUMA-560 has resulted in significant performance improvements [19, 21, 24]. The development of the application of this control algorithm with ARCADE_HAND on a finger of the UMDH is discussed next. This is followed by the implementation and evaluation of an AMBC algorithm for control of the UMDH.

4.2 Adaptive Model-Based Control Development

The general form of the output torque vector (τ) for a model-based control algorithm can be divided into feedforward (τ_{ff}), feedback (τ_{fb}), and auxiliary input (τ_{ax}) components.

$$\tau = \tau_{ff} + \tau_{fb} + \tau_{ax} \tag{4.1}$$

The feedforward torque of the adaptive model-based control (AMBC) in this thesis is defined as a matrix of known transcendental functions multiplied by a linear parameter vector:

$$\tau_{ff} = Y(q_d, \dot{q}_d, \ddot{q}_d)\hat{a} \tag{4.2}$$

The AMBC approach proposed by Slotine and Li uses parameter adaptation based on Lyapunov theory to compensate for model-based controller limitations [39]. The linear parameter vector which Slotine and Li proposed is given by

$$\hat{a} = \int \Gamma^{-1} Y^T(q, \dot{q}, \dot{q}_r, \ddot{q}_r)[(\dot{q}_d - \dot{q}) + \Lambda(q_d - q)] \tag{4.3}$$

where

$$\dot{q}_r = \dot{q}_d + \Lambda(q_d - q) \tag{4.4}$$

$$\ddot{q}_r = \ddot{q}_d + \Lambda(\dot{q}_d - \dot{q}). \tag{4.5}$$

Their proposal is slightly different from the approach actually implemented in the control algorithm being applied to the UMDH. The reason for the variation is that recent studies suggest that Slotine and Li's algorithm is unstable in the presence of velocity measurement noise [33, 35]. It so happens that the velocity data for the Utah/MIT Dexterous Hand is corrupted by spike laden position data from which the velocity is calculated. (See Equation 3.5.) Therefore, the variation proposed by Sadegh and Horowitz [33] on the passivity-based adaption approach, which eliminates the velocity measurement problem, is applied to the AMBC algorithm developed for the robotic finger. As a result of this variation, the regressor becomes strictly a function of the desired trajectory values and the adaption law is changed from that of Equation 4.3 to

$$\hat{a} = \int \Gamma^{-1} Y^T(q_d, \dot{q}_d, \ddot{q}_d)[(\dot{q}_d - \dot{q}) + \Lambda(q_d - q)]. \tag{4.6}$$

The Λ matrix is restrained by

$$\Lambda \leq \frac{K_D}{K_P}. \tag{4.7}$$

The feedback torque is given by

$$\tau_{fb} = K_D(\dot{q}_d - \dot{q}) + K_P(q_d - q) \tag{4.8}$$

and finally,

$$\tau_{ax} = 0. \tag{4.9}$$

The auxiliary torque (τ_{ax}) is introduced to compensate for the additional disturbances caused by using the desired values in the regressor. However, while maximum tracking performance may require both robust feedback and adaptive feedforward compensation, their effects must be separated for proper analysis of algorithm potential.

The regressor matrix (Y) is based on the known structure of the manipulator system dynamics. The manipulator platform for this thesis is joints 2 and 3 of finger 3 on the UMDH. These two joints form a planar elbow manipulator with revolute joints. The system dynamics and a regressor matrix for this configuration are presented in the tutorial by Ortega and Spong when they consider a planar manipulator with two revolute joints [34]. The regressor matrix is further enhanced by additional coulomb and static friction terms for each joint as developed by Leahy and Whalen [21].

The two finger UMDH regressor (Y) is implemented as a 2x13 matrix.

$$Y(q_d, \dot{q}_d, \ddot{q}_d) = \tag{4.10}$$

$$\begin{bmatrix} \ddot{q}_{d1} & \ddot{q}_{d2} & \ddot{q}_{d1} + \ddot{q}_{d2} & 2cosq_{d2}\ddot{q}_{d1} + cosq_{d2}\ddot{q}_{d2} - 2sinq_{d2}\dot{q}_{d1}\dot{q}_{d2} - sinq_{d2}\dot{q}_{d2}^2 \\ 0 & 0 & \ddot{q}_{d1} + \ddot{q}_{d2} & cosq_{d2}\ddot{q}_{d1} + sinq_{d2}\dot{q}_{d2}^2 \end{bmatrix}$$

$$\begin{bmatrix} \ddot{q}_{d1} & \ddot{q}_{d1}+\ddot{q}_{d2} & cosq_{d1} & cosq_{d1} & cos(q_{d1}+q_{d2}) & q_{d1} & 0 & sgnq_{d1} & 0 \\ 0 & \ddot{q}_{d1}+\ddot{q}_{d2} & 0 & 0 & cos(q_{d1}+q_{d2}) & 0 & q_{d2} & 0 & sgnq_{d2} \end{bmatrix}$$

where sign(x) is defined as:

$$sgn(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases} \tag{4.11}$$

The first six positions are related to inertial parameters. Gravitational forces are represented by parameters 6-8, and the final four parameters were added to account for viscous and coulomb friction. These three categories compose the thirteen diagonal positions of the regressor matrix which are employed in the feedforward calculation for the two joints.

The final step in the digital control system development was the implementation of the AMBC algorithm of equations onto a digital computer. The delay inherent in digital implementation was handled by using error information from the previous sample time in the current cycle output torque calculations. This changes the AMBC equations to the form

$$\tau_{ff}(k) = Y[q_d(k), \dot{q}_d(k), \ddot{q}_d(k)]\hat{a}(k) \tag{4.12}$$

$$\hat{a}(k) = \int_0^{T_s} \Gamma^{-1} Y^T [q_d(k), \dot{q}_d(k), \ddot{q}_d(k)][(\dot{e}(k-1) + \Lambda e(k-1)] \tag{4.13}$$

$$\dot{e}(k-1) = \dot{q}_d(k-1) - [q(k-1) - q(k-2)]/T_s \tag{4.14}$$

$$e(k-1) = q_d(k-1) - q(k-1) \tag{4.15}$$

$$\tau_{fb}(k) = K_D \dot{e}(k-1) + K_P e(k-1) \tag{4.16}$$

$$\tau_{ar}(k) = 0 \tag{4.17}$$

where T_s was the sample period of 333 Hz and the integration was accomplished using the Adams-Bashforth Two-Step method [5].

4.3 AMBC Implementation

Four main issues are involved in the successful implementation of the AMBC algorithm in the digital controller environment of the Utah/MIT Dexterous Hand. The most significant issue is tuning the adaptive controller. This was a time-consuming and painstaking process and is described in the first subsection. Subsequent subsections address the issues of parameter initialization and convergence.

4.3.1 Tuning the AMBC Controller AMBC tuning is a very heuristic process which is dependent on the manipulator, the number of adaptive parameters, and the individual components of the Γ^{-1} matrix. The simple selection of a diagonal Γ^{-1} matrix of common elements can result in improved performance or disaster. At a glance the procedure used to tune the Γ^{-1} diagonal parameters may seem very straight forward, but in reality this is not the case. Aggressively adapting certain parameters can cause instability. Also, the parameters are greatly interdependent. While a new value for a parameter may cause instability under one circumstance, that same value may enhance performance under different circumstances.

The tuning process was made even more difficult due to certain peculiarities of the tendon-driven manipulator. The primary obstacle was that when the manipulator is moved to the initial position by the PD controller, the existing tendon tensions upon completion of the movement are not accurately determined or preset. The difficulty is that these tendon tensions are part of the initial state of the test trajectory, resulting in variations of the initial state values of the manipulator at the start of a test trajectory. These variations sometimes resulted in unstable operation, or a run which was completely unordered. In other words, these runs were completely outside the normal training progression. Consequently, some operator discretion was

required to determine which training runs were unordered and to discard them from the training sequence. If the unordered runs are not discarded, then the dynamic parameter values resulting from such a run are at worst unstable and at best elongated the training period. Also, the training sequence was made somewhat indeterminate due to this tendency because the training runs have varied initial states, sometimes moving the dynamic parameters into an incorrect space. The first step taken in the AMBC tuning involved establishing the Γ^{-1} matrix diagonal for an established Λ and PD gains. Next, changes in Λ and the PD gains were conducted to determine their effects.

4.3.1.1 Tuning the Γ^{-1} Matrix Unlike previous studies in which the nominal values of the parameter vector were known a priori [24], parameter vector values are completely unknown for the fingers of the UMDH. Consequently, the Γ^{-1} matrix was tuned from an initial zero parameter state. Initial attempts to tune the Γ^{-1} matrix were conducted with values in the ranges previously generated by in-house AMBC trials on the PUMA 560 [21], but these values produced unstable AMBC performance. Therefore, the AMBC was run as closely to the baseline PD controller as possible, that is the feedforward torque was reduced almost to zero leaving the PD controller as the more significant contributor of the output torque. The diagonal adaptive gains of the regressor matrix were each "turned on" separately, starting from values as low as 10^{-5}. The general stability range was then established for each adaptive gain. The range went from as low as 10^{-4} to as high as 10^{-1}. The 13 adaptive gains were then separated into four groups, according to their dynamic representation. The first group consisted of the coupling inertial adaptive gains, 0-3. The next group was simply the inertial adaptive gains, 4 and 5, of each joint. Gravitational adaptive gains 6-8 composed the third group. The final group was the four friction representation adaptive gains, 9-12.

Each of the representative groups was independently "turned on" in the regressor matrix with each adaptive gain of the group assigned the value determined by

the individual tests. Again, adjustment of the Γ^{-1} values within each group was required as some groups became unstable or simply exhibited poor performance. Once the four groups were tuned, all 13 adaptive gains of the regressor matrix were activated together. Only one of the 13 adaptive gains was not necessary for maximum performance. Whenever the sixth value of the Γ^{-1} diagonal matrix was given a value it led to instability of the AMBC algorithm. Consequently, this value was set to zero and remained as zero in the final regressor matrix. With the full set of adaptive gains, the performance of the matrix was now acceptable. Lastly, each adaptive gain was increased and decreased as a member of the complete regressor matrix diagonal to determine if performance would be improved further. This resulted in two increases in the adaptive gains. The final set of Γ^{-1} diagonal values determined by this sequence of tests and utilized for subsequent evaluations was $(0.0002, 0.0001, 0.0001, 0.0001, 0.0001, 0.0, 0.001, 0.001, 0.001, 0.002, 0.01, 0.001, 0.01)$. The implementation of these values in the AMBC algorithm validated the AMBC concept for control of the UMDH with the tracking performance of Figure 4.1. The performance of Joint 2 with the AMBC controller is superior that of the PD controller. However, the Joint 3 performance actually degraded. The AMBC's inability to improve the Joint 3 tracking is believed to be a combination of two factors. First, the limited dynamic excitation in the trajectory for Joint 3 provides little for the AMBC adaptive mechanisms to learn. The second reason for the poor Joint 3 performance is the over-driving of the Joint 3 tensions. This second factor is discussed further as credence to this hypothesis unfolds.

A number of factors cause the training period of the AMBC algorithm on the robotic hand finger joints to be significantly longer than for a robotic manipulator such as the PUMA 560. These factors include the lack of nominal dynamics, the variations in the initial states, and the limited excitation of the trajectories which the UMDH is able to achieve. Additional tuning of groups and individual adaptive gains may continue to enhance algorithm performance, however, this was beyond

the scope of this thesis. The objective was the validation and initial evaluation of the AMBC concept on a tendon-driven manipulator for emulation of human finger movement, not to produce the best UMDH controller.

4.3.1.2 Lambda (Λ) Determination The value assigned to Λ determines the distribution of weight given to the position and velocity error in the $\hat{a}$ law of Equation 4.13. Consequently, changes in this value are expected to effect the AMBC algorithm's ability to track position and velocity. Tests were conducted to evaluate whether this hypothesis was true. The Λ matrix was set to a constant value approximating the ratio of position gain to velocity gain. For evaluation purposes the selected value was 100. The effects of variations in tracking performance were noted for a decrease in Λ from 100 to 50. Figures 4.2 and 4.3 demonstrate the effects of the change in Λ on the tracking error. As expected, the AMBC controller settles on the endpoint more quickly with $\Lambda = 100$ due to its emphasis on position error, while taking longer to adapt the peak tracking error. The smaller Λ demonstrates better initial tracking in the form of decreased peak errors as a result of the emphasis on velocity error but exhibits poorer endpoint performance. Also, the smaller Λ value improves the Joint 3 performance, thus supporting the hypothesis that Joint 3 is being over-driven. Note that the AMBC algorithm for $\Lambda = 100$ soon compensates for the initial lack of peak tracking error performance, and in the early training stages produces the better tracking error profile. The next test considers the long term effects of the change in Λ.

This test evaluates the ability of the AMBC control algorithm to compensate for changes in the weighting of the position and velocity, just as it did to unknown parameters in the dynamic model. The value for Λ has again been decreased to 50. However, for this test the trained parameters for $\Lambda = 100$ are used, and training is continued with the new value of Λ. Figure 4.4 reveals that although the tracking error of the new Λ degrades initially, by the ninth run the AMBC algorithm has compensated for the decrease in Λ and achieved similar tracking performance. The

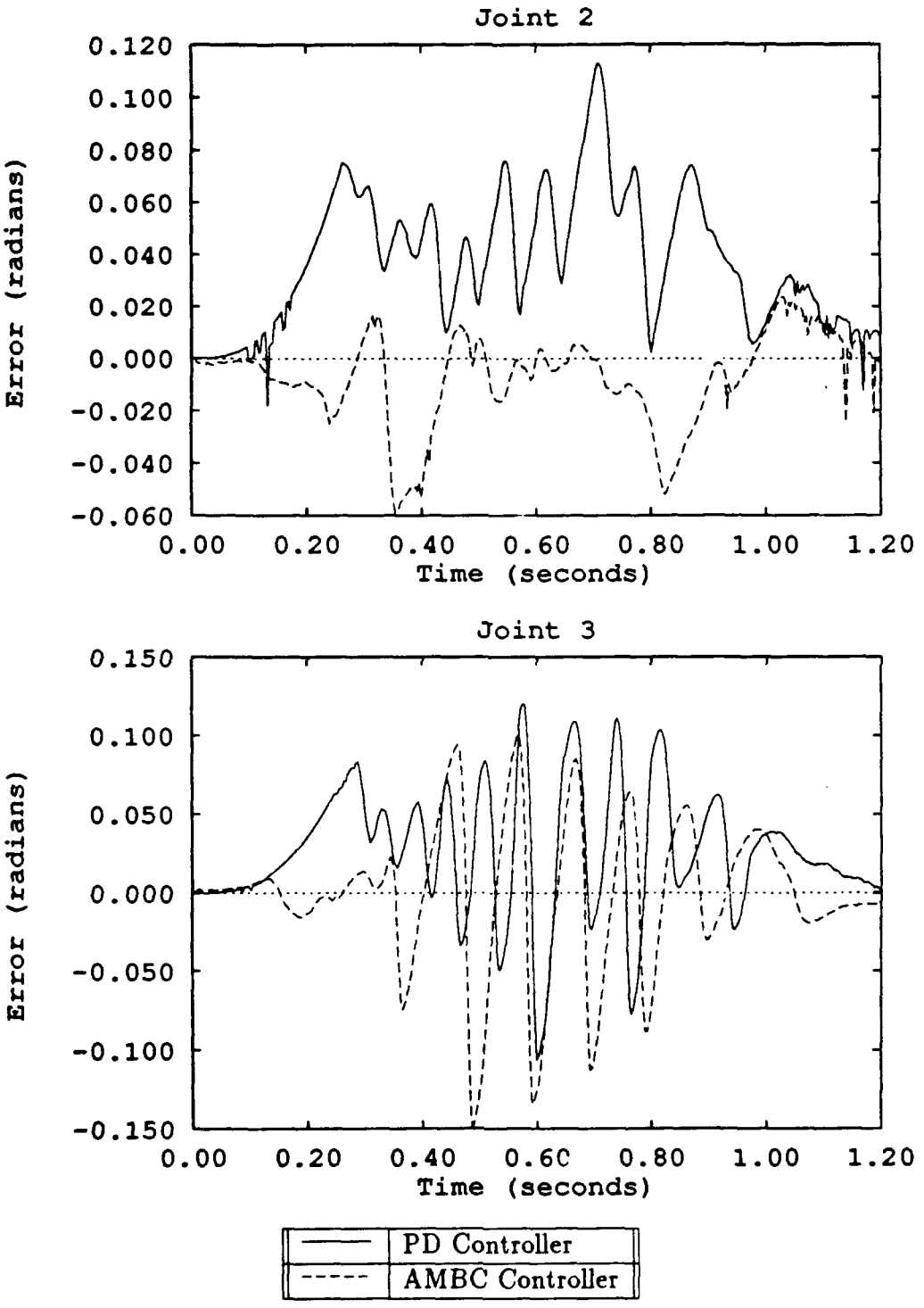

Figure 4.1. PD vs AMBC Tracking Error

value of Λ does not effect the long term tracking ability of the AMBC controller even though the higher value for Λ provides faster training.

4.3.1.3 Effect of PD Gain Variations The PD controller provides the feedback portion of the AMBC algorithm. Initial AMBC evaluations used PD gains identical to the PD baseline. Once the AMBC algorithm had been established, tests were conducted to determine the effects of variations in the PD gains on tracking. The PD gains in Table 3.2 were softened by at least one-third to the values recorded in Table 4.1. In general, reducing the feedback loop stiffness causes a degradation

Table 4.1. New Soft PD Feedback Gains

Link i	Position(K_P)	Velocity(K_D)
2	5.0	3.0
3	0.030	0.020

in tracking accuracy. However, the effects of the new PD gains are compensated for by the AMBC training mechanisms. Figure 4.5 shows the effect of the soft PD gains on the AMBC performance when using parameters that were trained with the higher PD gains. Although the tracking error for the initial training runs with the trained set of parameters and the reduced PD gains is initially worse than the new PD controller tracking, the performance quickly improves. The tracking performance of the sixth training run is superior to that of the stiff PD controller, and by the thirteenth run the tracking has improved by reducing peak tracking errors. These new PD gains with the AMBC algorithm are the second and final time that the oscillatory behavior of Joint 3 is consistently decreased, indicating that the original PD gains are too stiff for the sample rate. A training sequence was not conducted from $\hat{a} = 0$ for the softer PD gains because the manipulator became less stable and training too difficult.

Figure 4.6 compares the AMBC tracking with the soft gains and the stiff gains. The tracking with the soft gains is comparable to the stiff gain tracking, except for

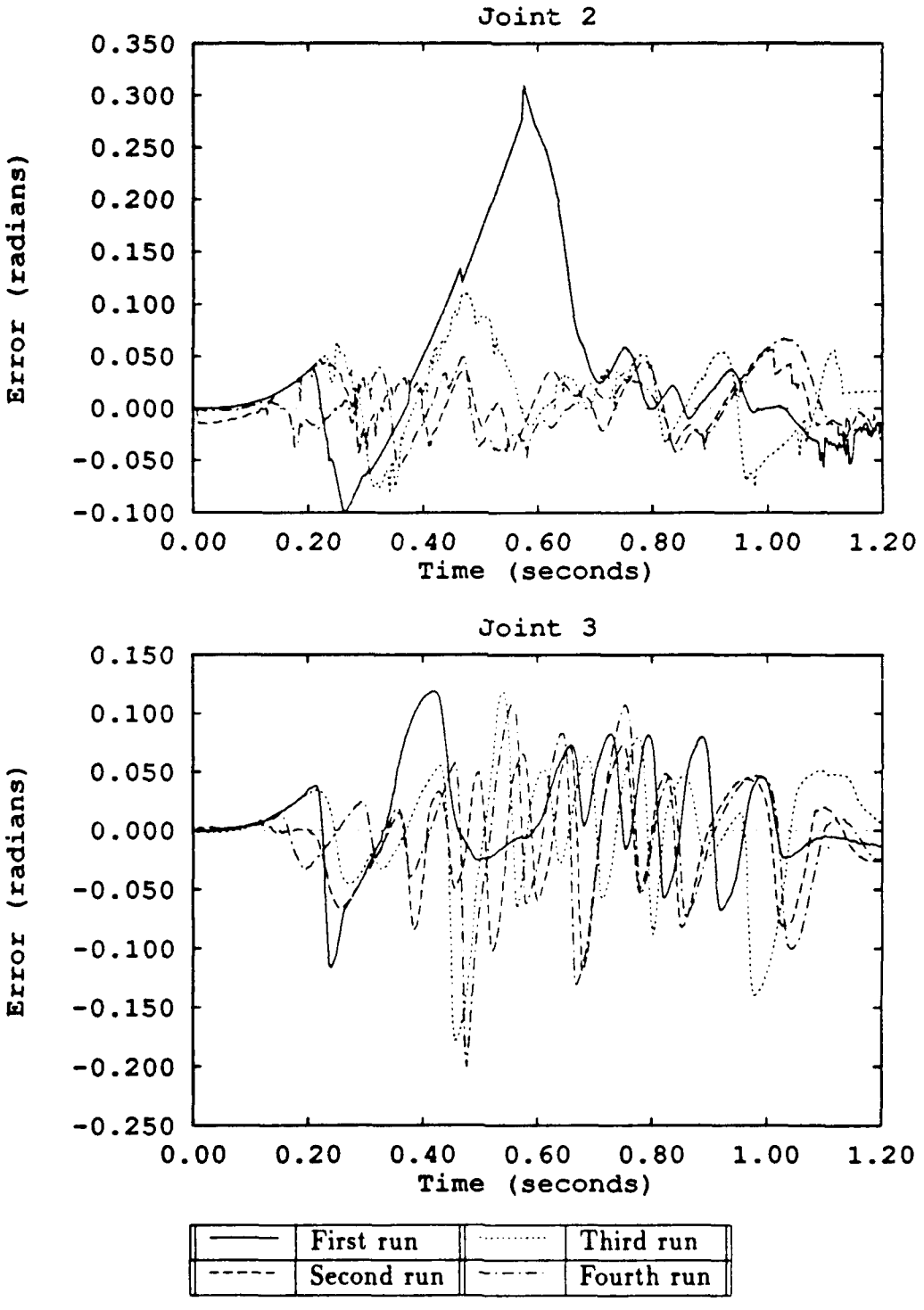

Figure 4.2. AMBC Tracking Error Adaption for $\Lambda = 100$

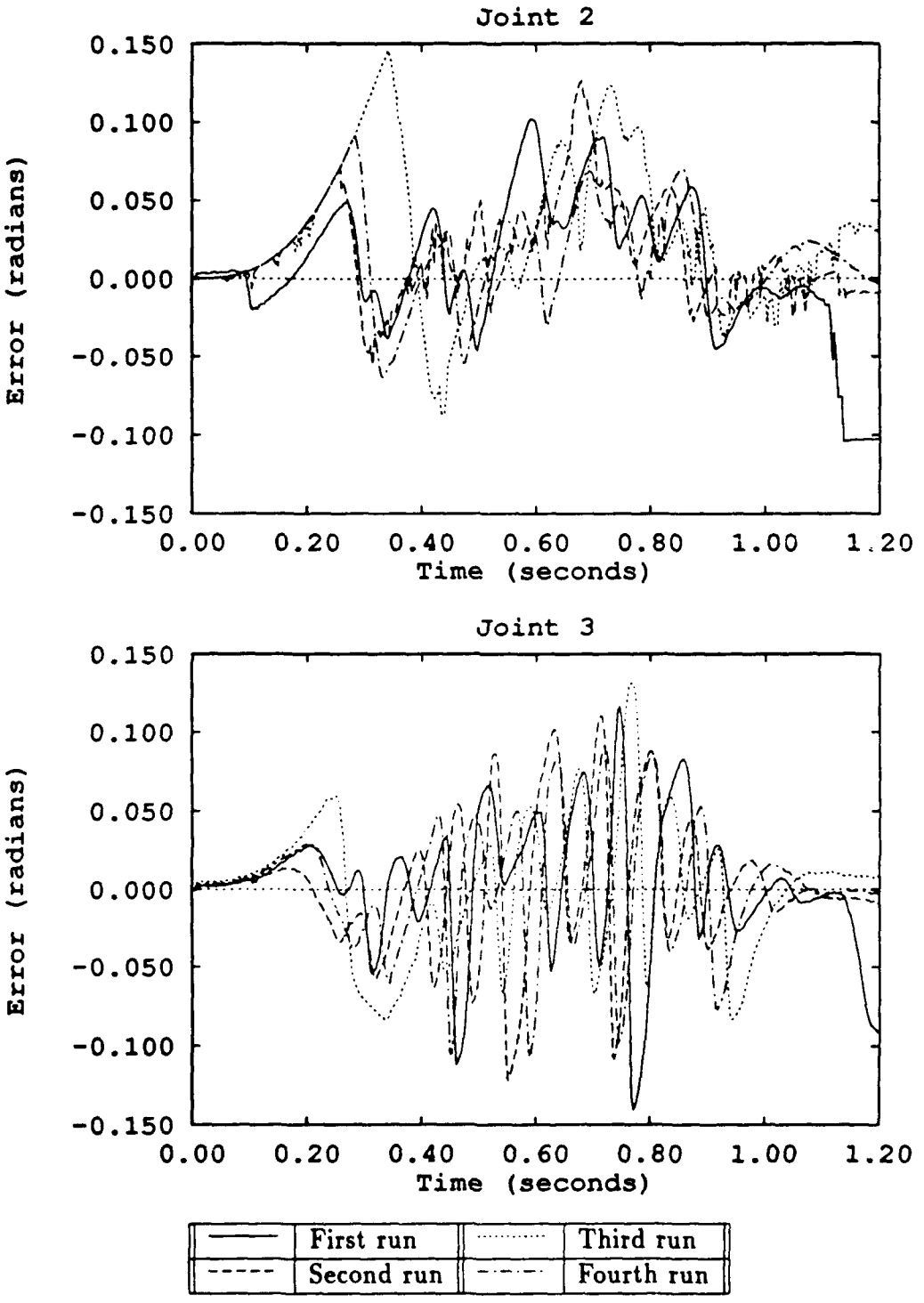

Figure 4.3. AMBC Tracking Error Adaption for $\Lambda = 50$

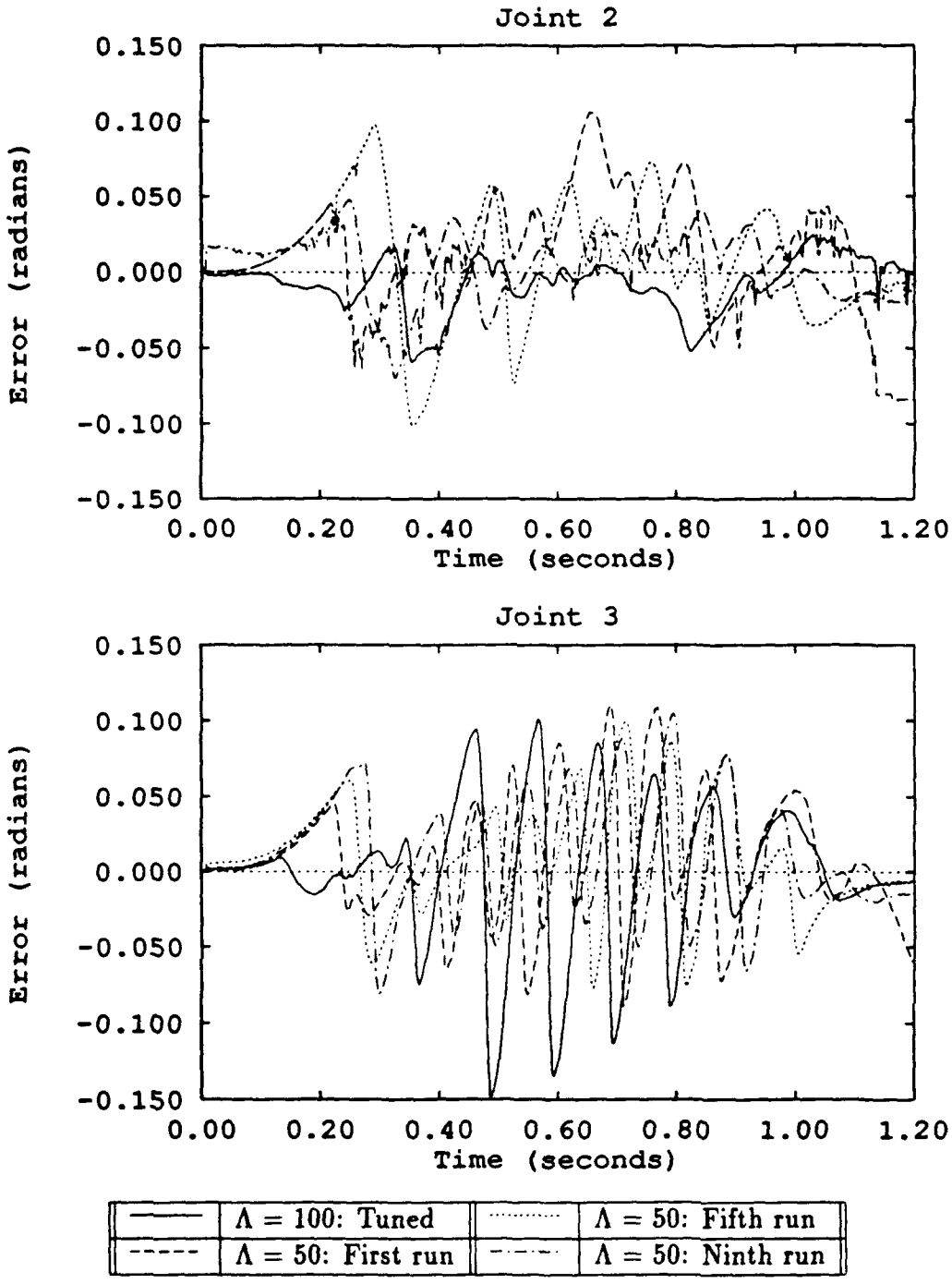

Figure 4.4. Effect of Λ on Tracking Error

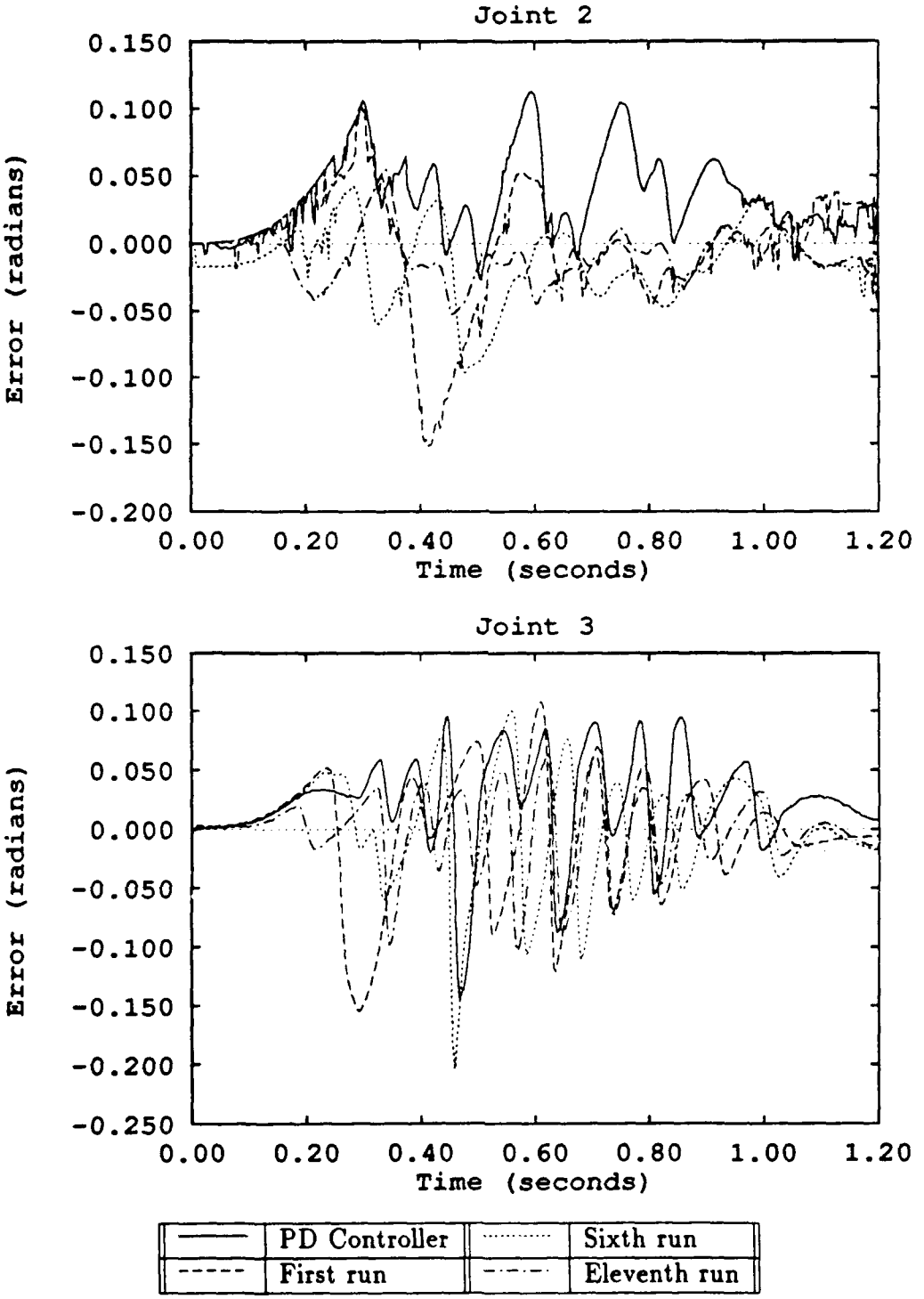

Figure 4.5. AMBC Tracking Error with New Soft PD Gains

a small loss in endpoint performance. The AMBC algorithm removes the stiff gain requirement for good tracking, which also removes the oscillation previously seen for high gain implementations.

4.3.2 Parameter Initialization The transient tracking performance is directly dependent on the a priori knowledge of the parameter vector. In previous AMBC studies conducted on the PUMA 560 [21] the parameters were separated into both known and unknown dynamics, a_n and $\hat{a}$. Although nominal dynamics aide in AMBC performance, the algorithm has the capability to adjust the parameters to compensate for the disturbances of a reduced dynamic model. In the UMDH test case there is no dynamic model, thus the adapted parameters must compensate for the complete dynamic model. The parameters represented in Equation 4.13 by $\hat{a}$ are thus initially set to zero, reducing the AMBC to a pure feedback controller at the start of the evaluation. Figure 4.7 reveals the progressive success that can be achieved in tracking performance even when all the dynamic parameters are initialized to zero. The lack of dynamics for Joint 3 to learn limited its ability for any real improvement.

4.3.3 Parameter Convergence AMBC has the theoretical ability to learn the actual dynamic coefficients for the manipulator dynamics, given a trajectory with persistent excitation [21]. Final parameter values are dependent on both initialization and adaptation gains. When the a priori parameter knowledge is non-existent, as in this application on the robotic hand, the Γ^{-1} influence is significant. The closer the Γ^{-1} matrix reflects the relative magnitude of the actual parameters, the closer the parameters will reflect the physical values.

Several factors contribute to the extended training period of the UMDH. The lack of nominal parameters results in a longer adaptation period for the robotic hand. The variations inherent in the initial states of the training runs often contribute to both longer training sequences and an indeterminate number of required training runs per training sequence. Another contributor to the deficiency in training the

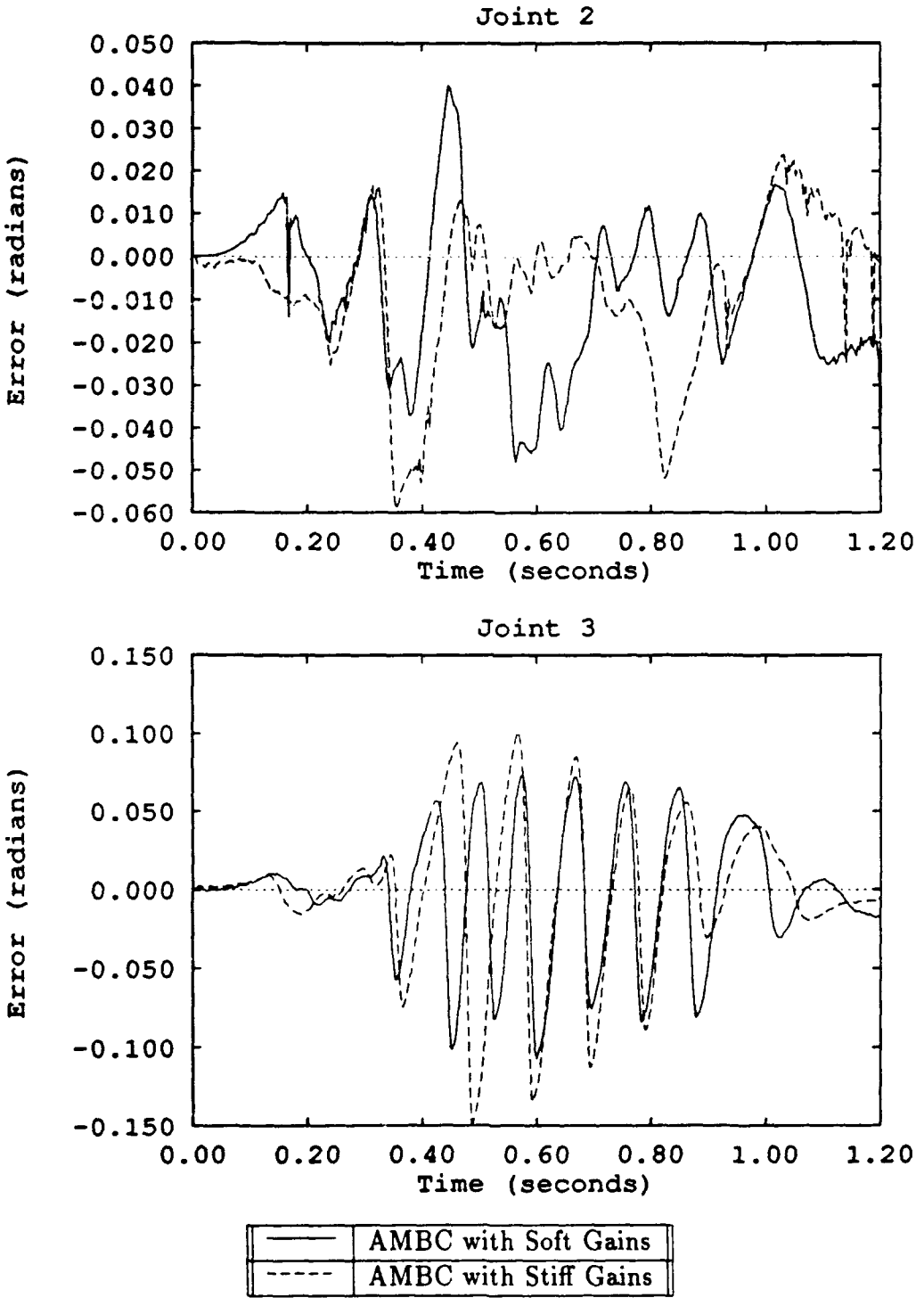

Figure 4.6. AMBC Tracking Error with Soft and Stiff PD Gains

parameters on the hand is the limited dynamic excitation achieved with the trajectories which are typical of human finger movement. The trajectories on a PUMA 560 can be designed for peak parameter excitation. On the UMDH the trajectories are constrained by the extension and flexion of the finger joints. These movements lack the rotational tendencies of an industrial manipulator, and the joints typically either extend or flex in unison. The AMBC algorithm needs 10 to 20 training runs for the UMDH joints, where as the PUMA 560 requires only 4 to 6 training runs [24]. Figure 4.7 reveals the adaptation and learning capabilities of the AMBC approach for the two joints of the UMDH finger with no nominal parameters. Note that training is still progressing even after 20^{+} iterations. The slight degradation in Joint 3 performance throughout the training progression again indicates the lack of dynamic excitation from which it is able to learn.

Parameters can be arranged such that different sets produce identical profiles. The adaptation mechanism does not learn the actual physical values, but responds to the effect of the parameters on the tracking error. Two sequences of multiple runs, each with no nominal parameters, will produce different sets of parameters and varied tracking error profiles. Figure 4.8 includes the best-case runs from two different adaptation sequences which started from ground zero with identical Γ^{-1} matrix values and $\Lambda = 100$. Even though the endpoint and peak tracking errors are very similar, the error profiles of the two sequences vary greatly. The adaptation parameters also vary significantly. Table 4.2 compares the starting and finishing adaptation parameters of both best-case runs. The magnitudes of the sequence 2 parameters are significantly higher than those of sequence 1, with the greatest disparity being the sign difference of the twelfth parameter.

Once training has achieved the best-case set of dynamic parameters, one may be inclined to discontinue the adaptation mechanisms. However, this line of thinking is flawed. Even though the parameters have been trained to a path in the parameter space which produces the best tracking, the adaptation mechanisms continue to

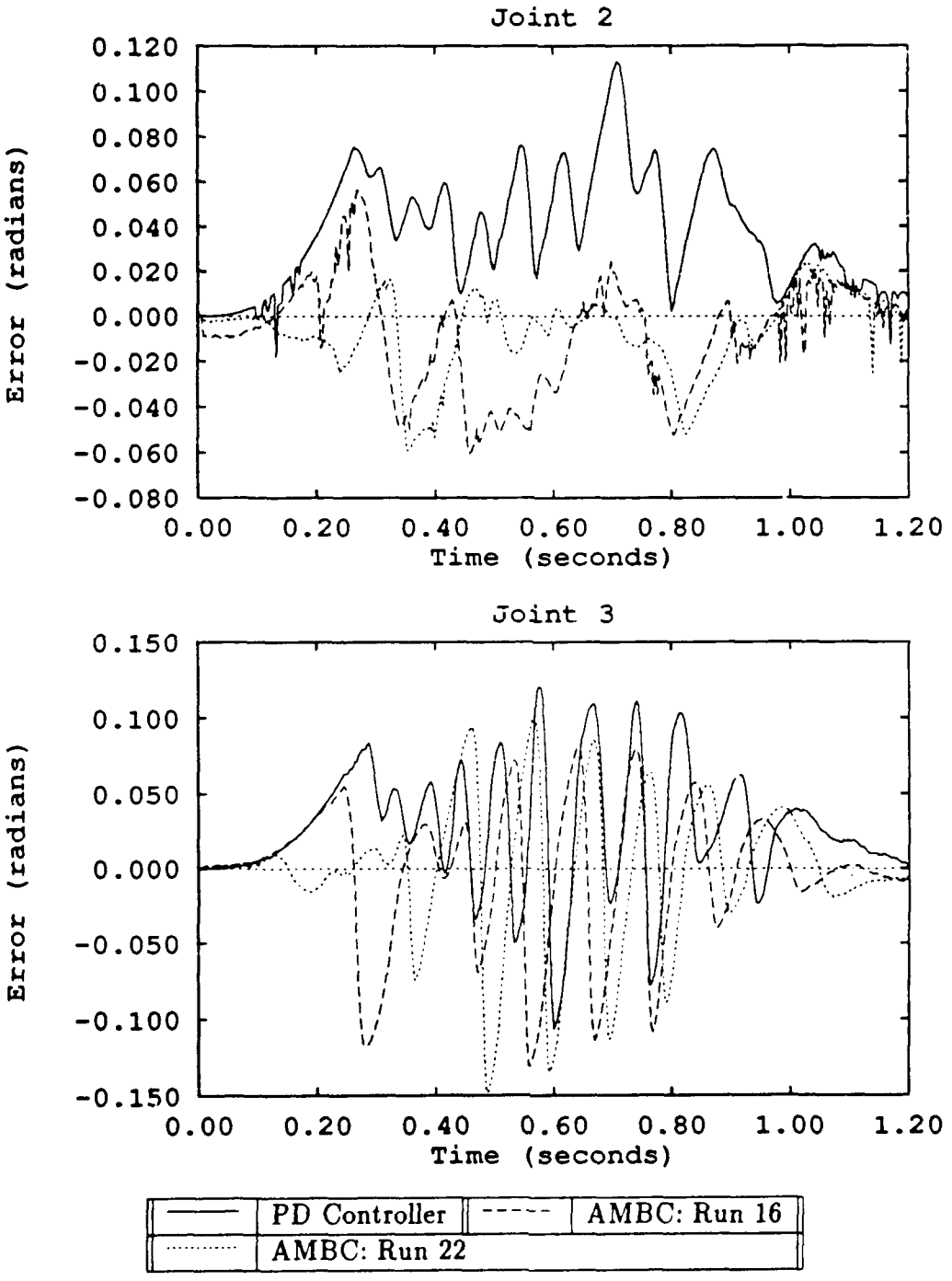

Figure 4.7. AMBC Tracking Error: No Initialization

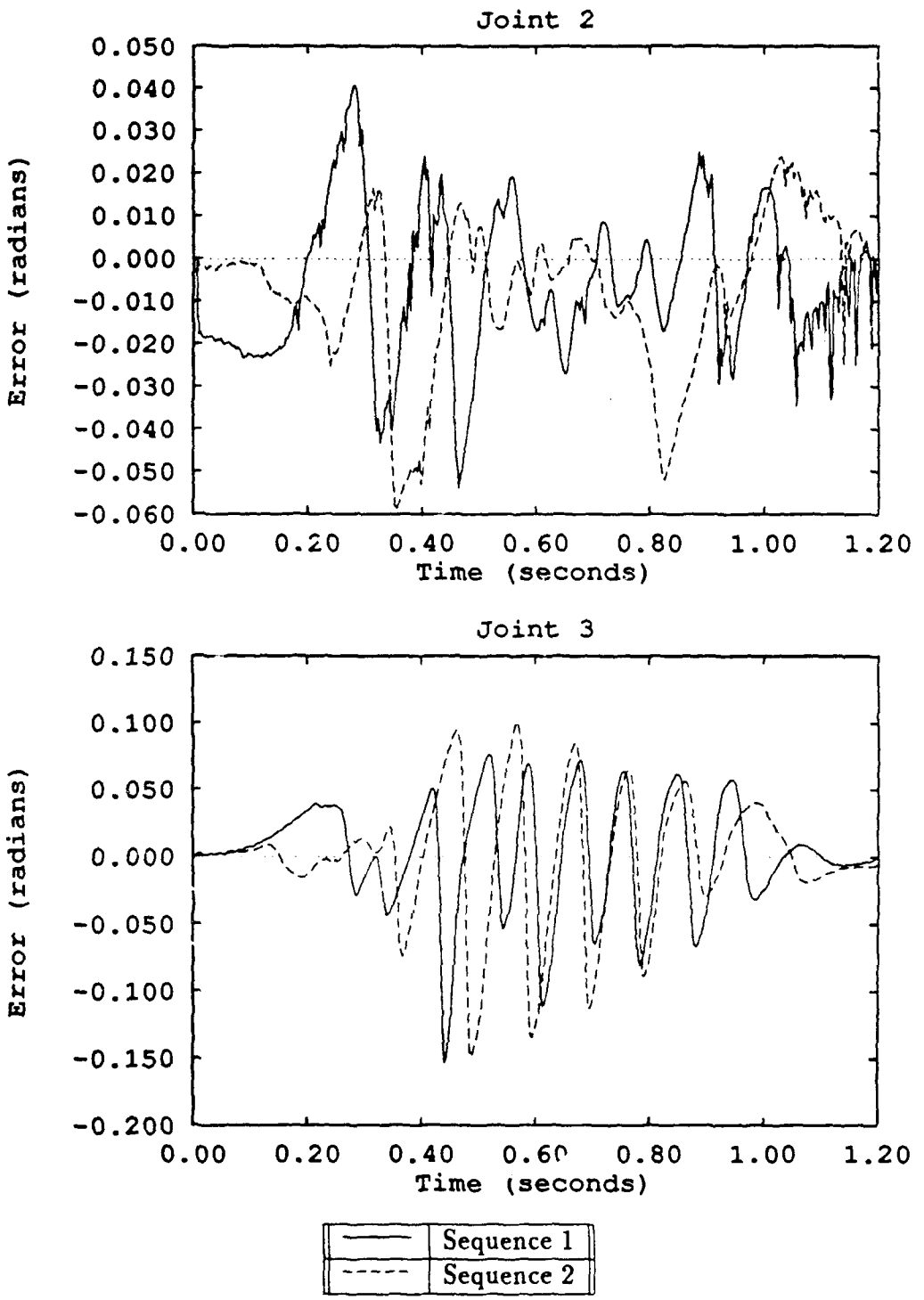

Figure 4.8. AMBC Best-Case Sequence Comparison

adapt to the dynamics throughout the trajectory. The dynamic parameters do not remain constant throughout the trajectory. Evidence of this is given in Table 4.2 which shows how the parameters continue to change from the start to the comple-

Table 4.2. Adaptation Parameter Comparison

Pos.	Run 1: Start	Run 1: Finish	Run 12: Start	Run 12: Finish
0	0.0043	0.0033	0.0109	0.0089
1	0.0020	0.0015	0.0055	0.0045
2	0.0007	0.0014	0.0061	0.0024
3	-0.0035	-0.038	-0.0105	-0.0130
4	0.0020	0.0015	0.0055	0.0045
5	0.0000	0.0000	0.0000	0.0000
6	0.0145	0.0139	0.0209	0.0201
7	0.0145	0.0139	0.0209	0.0201
8	0.0089	0.0088	0.0136	0.0128
9	0.1112	0.1080	0.2020	0.1950
10	0.0444	0.0456	0.0744	0.0656
11	0.0181	0.0174	0.0256	0.0247
12	-0.0234	-0.0208	0.0176	0.0161

tion of a run. Figure 4.9 reveals what happens when the adaptation mechanism is "turned off" when executing a trajectory. The parameter values implemented are those from sequence 2 of Table 4.2. Whether the parameter values used are start or finish values of the best-case trajectory does not help the tracking performance if the adaptation is stopped. As the manipulator proceeds through the desired path, the lack of adaptation results in significant tracking degradation. The degradation of tracking for the UMDH joints due to "turning off" the adaptation mechanism was more significant than for the PUMA 560 [24, 21]. Once optimum dynamic parameters have been achieved, that is the start values of the best tracking performance obtained, these values should be repeatedly used to initialize the parameters for future executions of the trajectory. Further training should be discontinued, however, the adaptation mechanisms should remain on for execution of the trajectory.

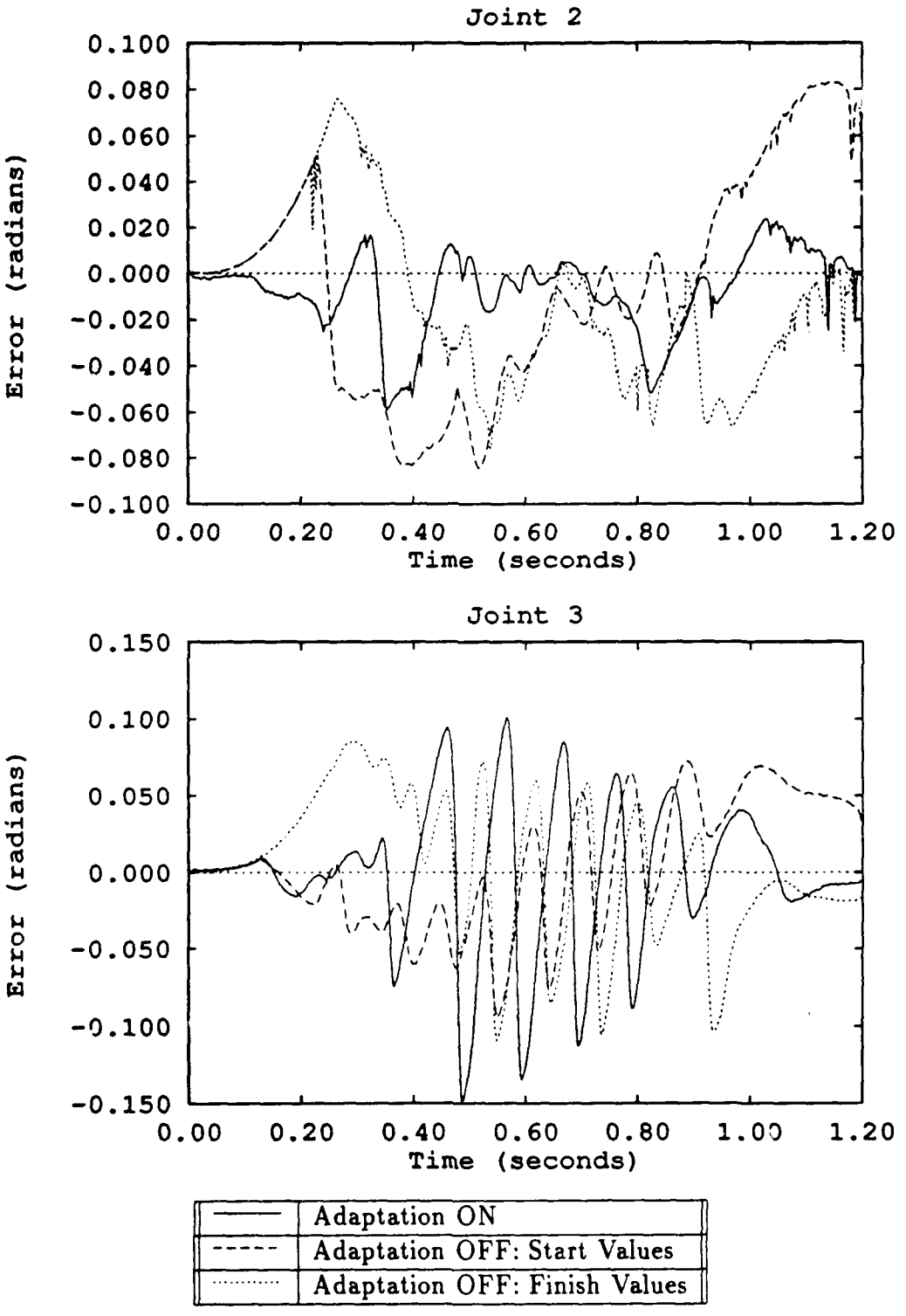

Figure 4.9. AMBC Tracking with With and Without Adaptation

4.4 Experimental Performance Evaluation

The final step in the digital controller development is the performance evaluation of the AMBC algorithm. The evaluation must consider whether the algorithm meets the following tracking requirements mentioned in Chapter 3: compensation for robotic dynamics by adaption/learning of model uncertainty, multiple trajectory tracking capabilities, and payload adaptation. The first issue addressed is the ability of the AMBC algorithm to adapt to model uncertainty, and the tendencies of the adaptation mechanism. The second issue of concern is whether the AMBC algorithm can adapt to multiple trajectory tracking. Can the parameters which evolve from training on one trajectory aide in the tracking of a different trajectory? Finally, tests are conducted to determine the AMBC algorithm tendencies when unknown payloads are implemented.

4.4.1 Adaption/Learning of Model Uncertainty Application of AMBC algorithms on platforms such as the PUMA 560 allow for the inclusion of nominal parameters which provide the algorithm with an initial set of parameters from which to train [21]. As discussed in the subsection on Parameter Convergence, a number of factors cause the training period of the AMBC algorithm on the robotic hand finger joints to be significantly longer than for a robotic manipulator such as the PUMA 560. Despite these less than optimal conditions, the AMBC experimental evaluations indicate that feedforward adaptation provides significant compensation for the manipulator when nominal dynamics are unknown.

In Figure 4.7 the tracking performance of the AMBC algorithm on the UMDH is compared to the tracking performance of the PD controller which composes the baseline of the digital controller. The initial run results in a very high joint 2 peak error. This single high peak error in the first run occurs consistently within the first two training runs of the AMBC algorithm. By the second run, the tracking error is improved over that of the PD controller. A look at subsequent plots would reveal the

continued adjustments of the AMBC mechanisms as the tracking error continues to fluctuate as in the third run. The fluctuations gradually diminish as the manipulator becomes more stable and tracking error decreases. However, despite the training progress intermittent large tracking errors continue to occur. This is indicative of the non-asymptotic training progression of the AMBC algorithm. Similar results have been previously reported for the PUMA 560 [24, 21] The best-case tracking profiles are achieved after twenty to thirty runs. Figure 4.10 shows the training progression of selected AMBC training runs en route to the best-case profile.

4.4.2 Multiple Trajectory Tracking Once the AMBC algorithm was trained for the test trajectory, a new test trajectory was implemented to determine the algorithm's ability to track multiple trajectories. Tests were performed to determine the response of the AMBC algorithm to the new trajectory. More specifically, the tests consider the transportability of the Γ^{-1} regressor matrix which was tuned for the initial test trajectory and the adaptability of the dynamic parameters which resulted from the trial runs of the initial trajectory. The change instituted for the new test trajectory is that the palm of the UMDH is placed facing upwards. The position, velocity, and acceleration trajectory profiles of the joint remain the same. Consequently, joint motion is no longer across the gravitational field. Joint 2 now moves directly into the gravity field, wh Joint 3 moves in and out of the gravity field.

Initially, a set of training runs was made from ground zero using the Γ^{-1} regressor matrix diagonal values tuned for the first trajectory. In Figure 4.11 two tracking error profiles of the AMBC runs are presented and compared to the PD controller performance for the new trajectory. The AMBC algorithm was able to track the new trajectory and outperform both the PD controller and the AMBC performance on the first trajectory. The improved AMBC performance on the new trajectory may result from the increased dynamic excitation of the new trajectory into the gravity field. Also, the AMBC tracking of the new trajectory indicates that the Γ^{-1}

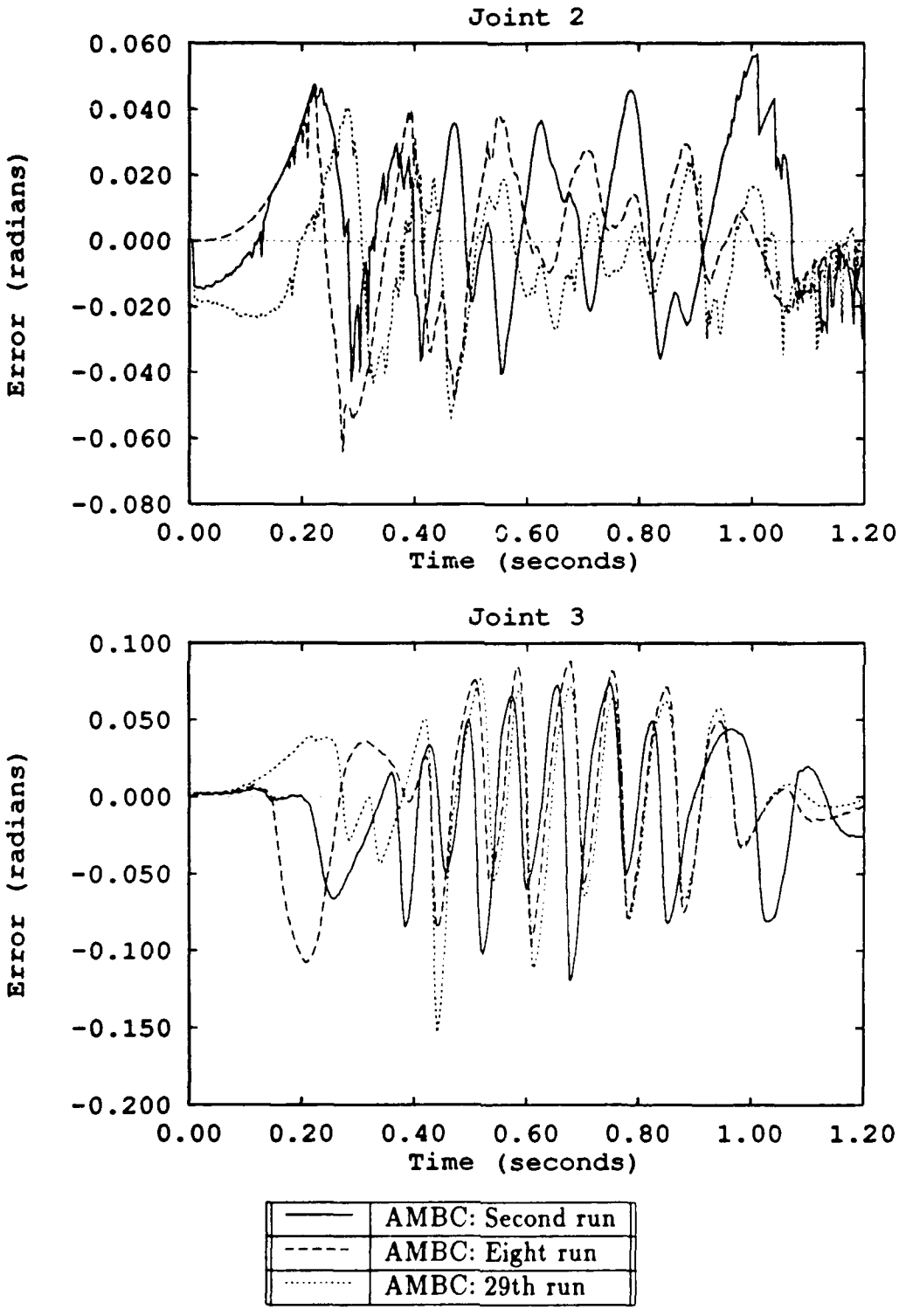

Figure 4.10. AMBC Training Progression

matrix is transferable to other trajectories. Figure 4.12 compares the AMBC tracking performance for the two trajectories, both trained from $\hat{a} = 0$. Now that the transportability has been determined, the next issue is the adaptability of the dynamic parameters trained from the first trajectory. In theory these parameters go to the correct $\hat{a}$ values of the manipulator and are trajectory independent. In practice these parameters migrate to the specific physical dynamics of the trajectory on which they are trained. The results of AMBC training with the Γ^{-1} matrix diagonal values tuned for the first trajectory, and the dynamic parameters resulting from training with the first trajectory are shown in Figure 4.13. Training runs were conducted and are compared to the PD controller performance. The dynamic parameters are not directly adaptable from one trajectory to the other, but the tracking is still stable and is only slightly worse than the PD tracking. Just as the AMBC adaptation mechanism is able to compensate for unknown dynamics, the mechanism adapts to the change in dynamics caused by the new trajectory in the same way.

4.4.3 Payload Adaptation Finally, tests are conducted to determine the adaptability of the AMBC algorithm to an unknown payload on the robotic finger. The payload consists of a Lincoln penny taped to the underside of Link 2. The original test trajectory is executed for the tests. Figure 4.14 shows the ability of the AMBC algorithm to compensate for the unknown payload with no a priori knowledge of the manipulator dynamics. The training is initiated from ground zero, that is $\hat{a} = 0$, and the tracking performance is soon improved over that of the PD controller, just as it was without the payload. Figure 4.15 highlights the comparison of the AMBC vs. PD tracking variations due to payload. The AMBC training mechanisms are also able to provide the same caliber of tracking performance with or without the payload. On the other hand, the initial tracking of the PD controller degraded with the addition of the payload.

The next test examined whether the dynamic parameters achieved by training without a payload improve tracking with a payload. The trained set of parameters

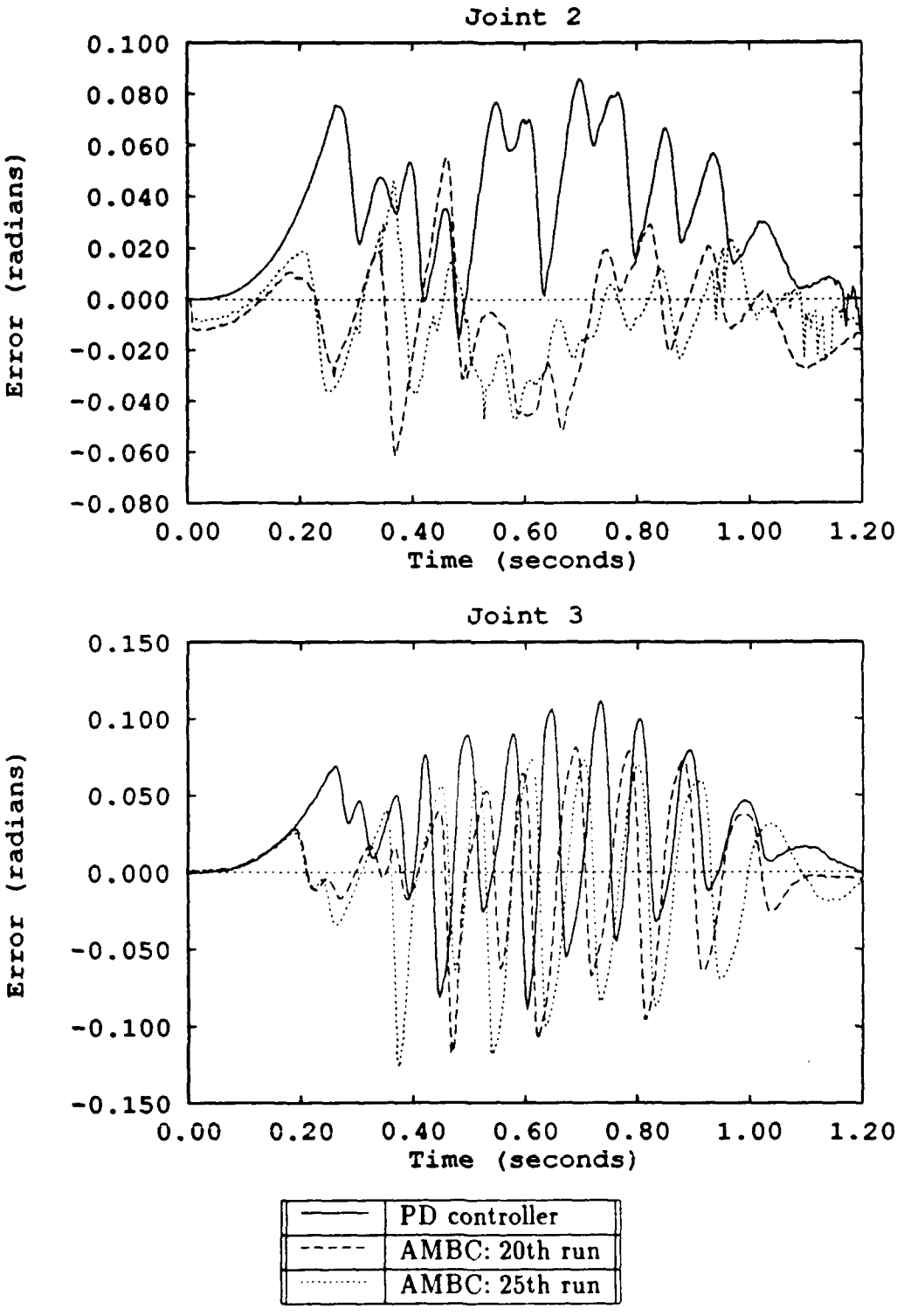

Figure 4.11. AMBC Performance: New Trajectory

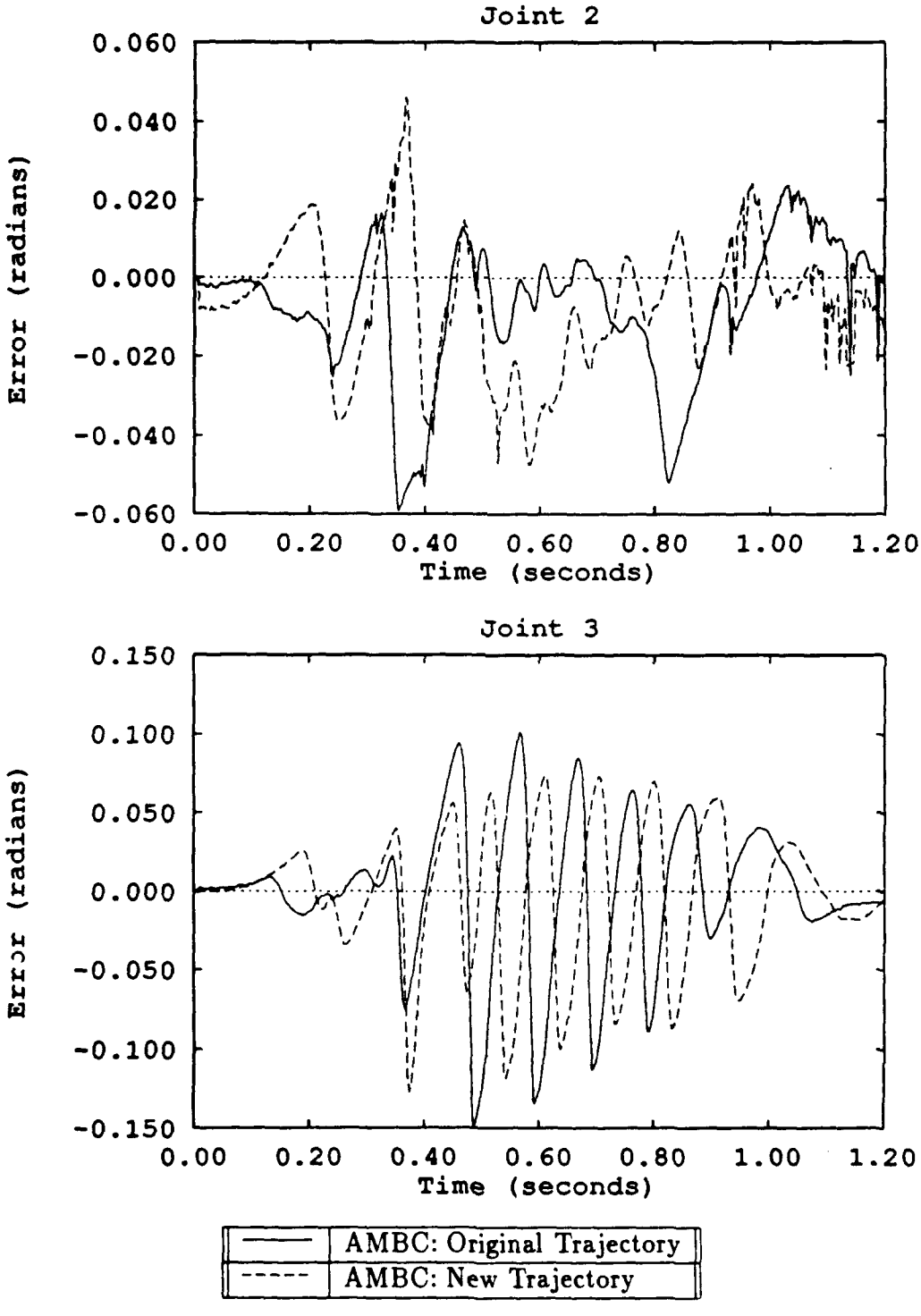

Figure 4.12. AMBC Tracking for Two Trajectories

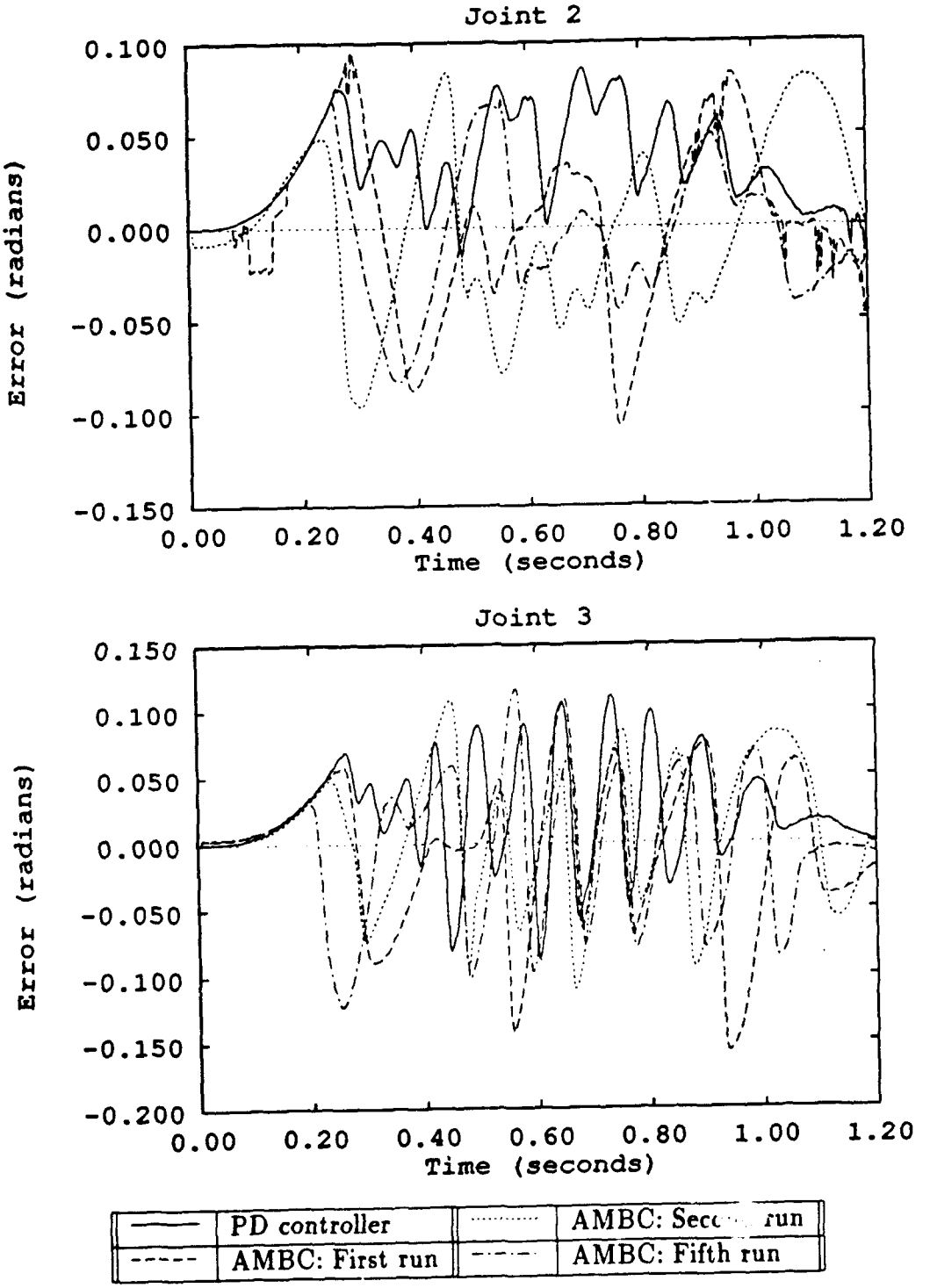

Figure 4.13. AMBC Performance: New Trajectory with Old Parameters

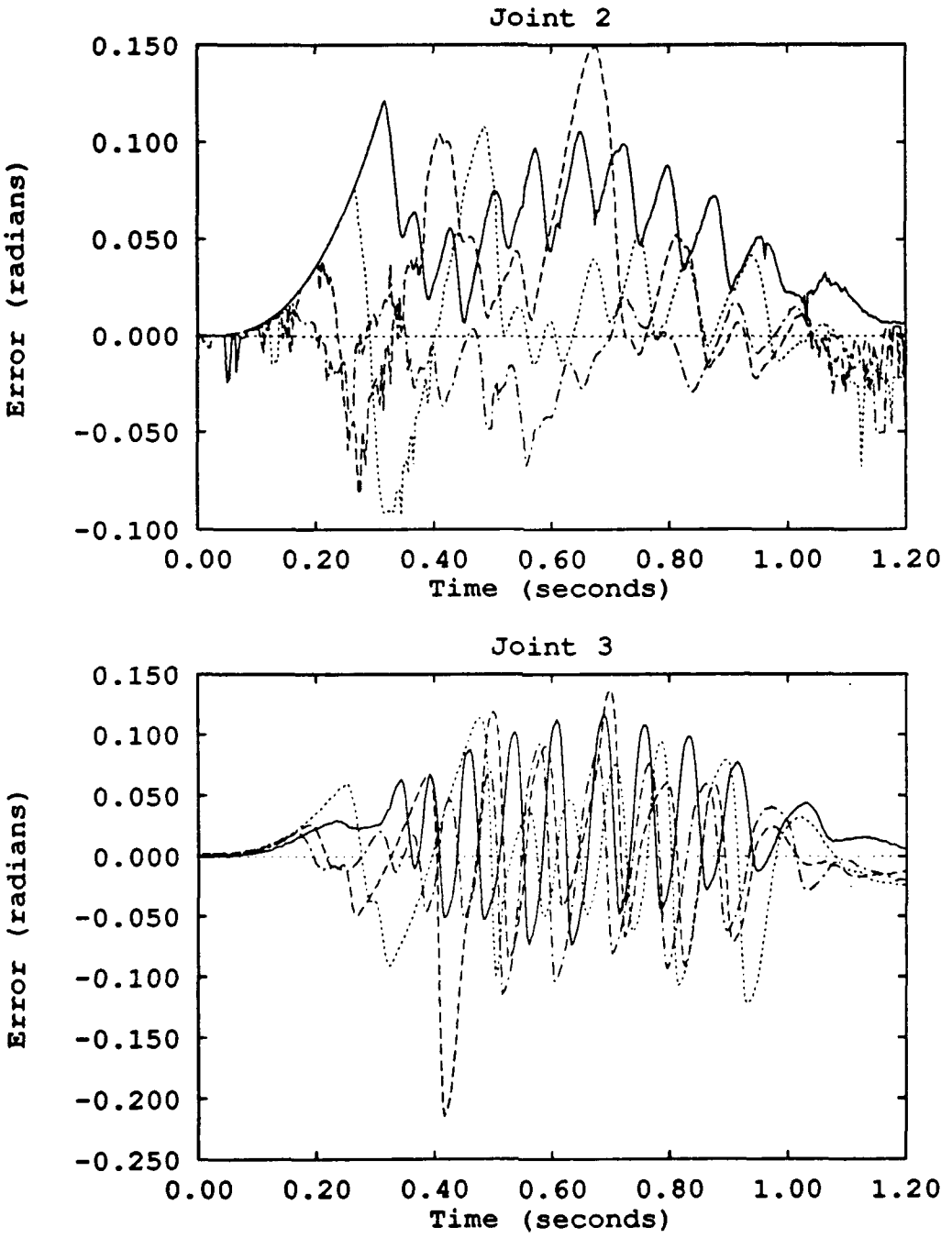

Figure 4.14. AMBC Tracking Error with Payload

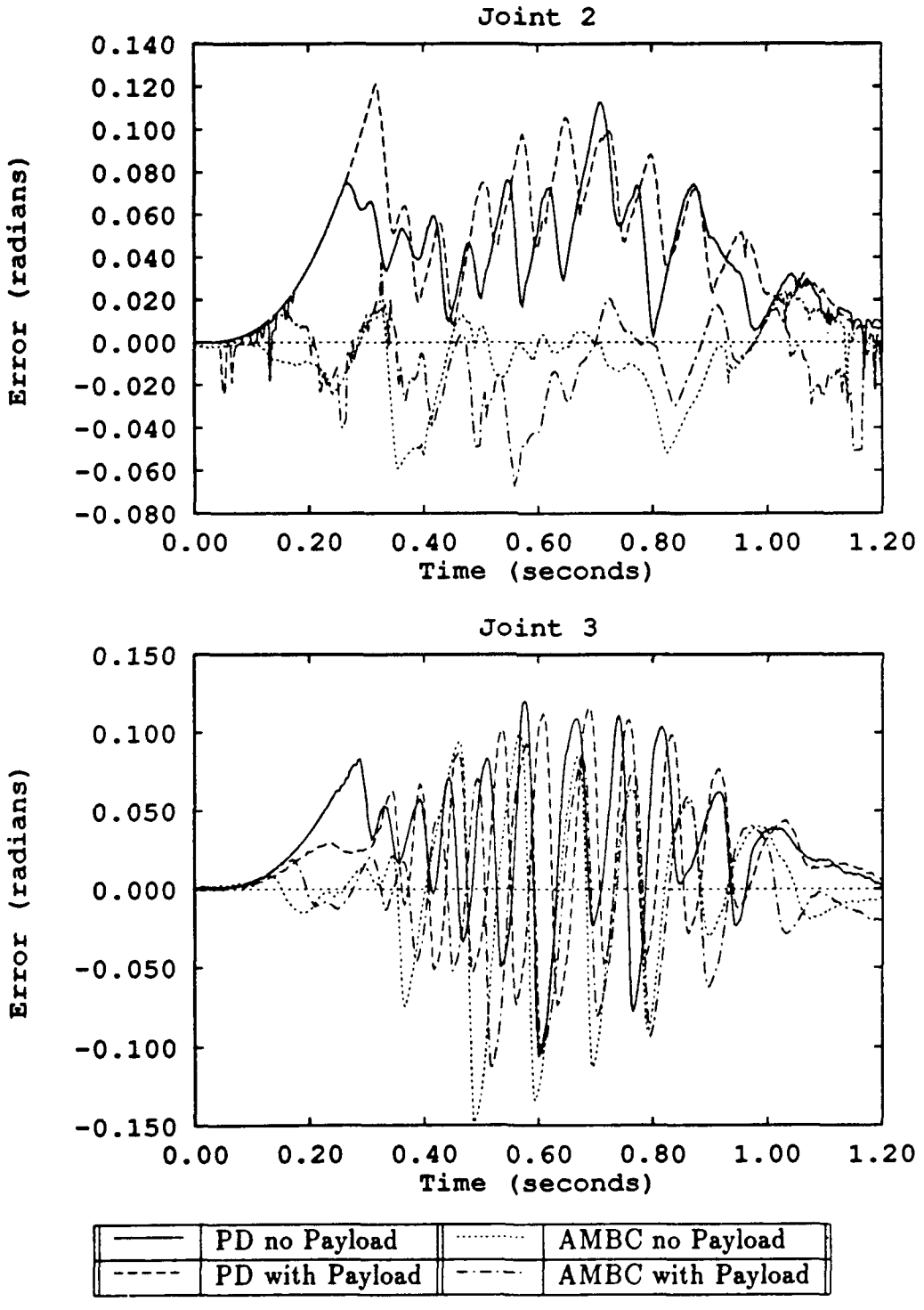

Figure 4.15. PD vs. AMBC Payload Tracking

are used to initialize the AMBC algorithm and new training runs are executed with the unknown payload. Figure 4.16 shows that from the very first training run, the AMBC algorithm with the dynamic parameters from the no payload case tracks better than the PD controller. Also, the dynamic parameters quickly retrain for the dynamics of the unknown payload and enhance the tracking performance almost threefold. Table 4.3 compares the parameter values at the start and finish of the first and last training runs with the payload. Initializing the AMBC algorithm with the

Table 4.3. Adaptation Parameter Comparison: Payload

Pos.	Run 1: Start	Run 1: Finish	Run 12: Start	Run 12: Finish
0	0.0043	0.0056	-0.0005	-0.0002
1	0.0020	0.0029	-0.0002	-0.0000
2	0.0007	-0.0002	0.0000	0.0002
3	-0.0035	-0.0128	-0.0011	0.0004
4	0.0020	0.0029	-0.0002	-0.0000
5	0.0000	0.0000	0.0000	0.0000
6	0.0145	0.0199	0.0149	0.0142
7	0.0145	0.0199	0.0149	0.0142
8	0.0089	0.0123	0.0113	0.0112
9	0.1112	0.1880	0.1375	0.1312
10	0.0444	0.0578	0.0373	0.0280
11	0.0181	0.0248	0.0186	0.0177
12	-0.0234	0.0138	0.0172	0.0152

previously trained no-payload parameters improved the stability of the robotic finger and allows faster adaptation to the unknown payload than when the parameters are initialized to zero. The difference between tracking accuracy with and without payload is negligible after learning. This was the same conclusion reached for the slower trajectory on the PUMA 560 [21].

4.5 Summary

The control requirements for the Utah/MIT Dexterous Hand were discussed and an AMBC algorithm was selected as the initial control method evaluated to meet

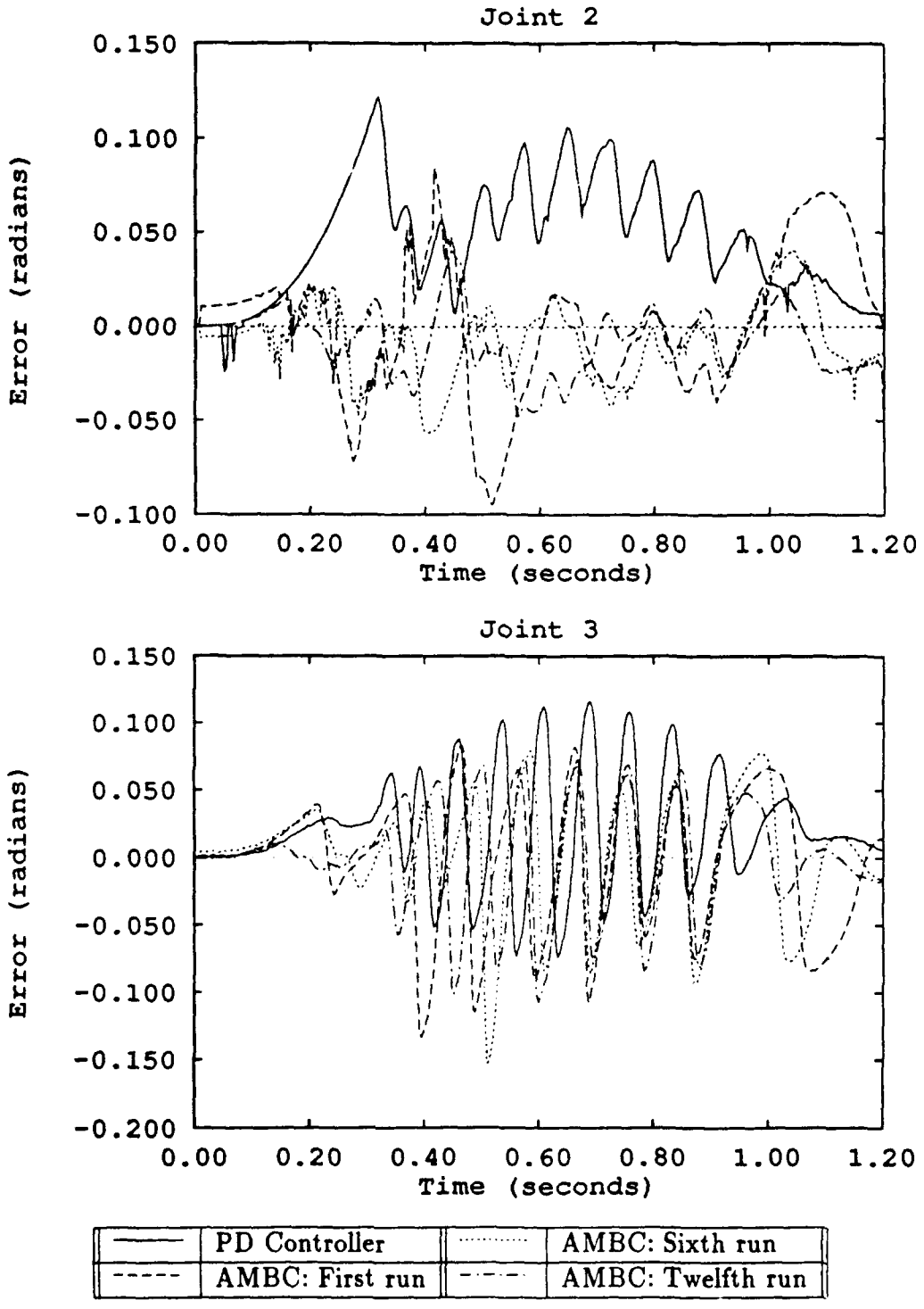

Figure 4.16. AMBC Tracking with Payload and Parameters Trained with No-Payload

the requirements. An AMBC algorithm was developed for the gross motion control of two proximal joints of one finger of the UMDH. The foundation of the AMBC algorithm is the approach proposed by Slotine and Li [39], with the variations suggested by Sadegh and Horowitz [33]. The AMBC algorithm was coded for evaluation under the ARCADE_HAND environment. This is the first known application of an AMBC algorithm on a pneumatic tendon-driven manipulator. The AMBC implementation issues of tuning, parameter initialization, and parameter convergence are discussed as they pertain to operation of the UMDH. An experimental evaluation of the AMBC algorithm performance on the UMDH robotic manipulator was conducted and comparisons were made to previous results obtained on the PUMA 560. The evaluation tests revealed that the AMBC algorithm is able to compensate for dynamic model uncertainty, provide multiple trajectory tracking, and adapt to unknown payloads. Indications are that the range of applications of AMBC can be extended to manipulators whose dynamics are dominated by drive system forces.

V. Conclusions and Recommendations

5.1 Summary

A digital control environment, ARCADE_HAND, was proposed, developed, and implemented for the Utah/MIT Dexterous Hand (UMDH). The baseline controller for the digital control system is a Position-plus-Derivative (PD) feedback control loop. This controller was developed and evaluated for applicability to the tendon-driven UMDH. Once the new digital test platform was completed and the PD controller established, an alternative control algorithm was developed and evaluated. Adaptive Model-Based Control (AMBC) was determined to be a promising prospect for control of the UMDH for human movement emulation. Previous implementations of AMBC were discussed and the theory behind AMBC was presented. Manipulator specific variations were made to previous implementations of AMBC to tailor the algorithm for operation in the ARCADE_HAND environment. The AMBC algorithm was tuned for the UMDH and tests were conducted to determine the performance characteristics. Additional tests evaluated the ability of the AMBC to meet the requirements needed for emulation of human finger movement.

5.2 Conclusions

The ARCADE_HAND experimental environment proved suitable for dexterous control and manipulation studies on the UMDH. Evaluation of the baseline digital controller of this environment found the PD controller to be lacking important attributes required for dexterous motion and manipulation studies. The lack of compensation for unknown dynamics which is needed for high speed tracking and payload invariance led to the development and evaluation of AMBC as an alternative control algorithm for emulation of human finger movement. The regressor matrix, Γ^{-1}, of the AMBC algorithm was tuned for the test trajectory and provided significant tracking improvements after training. Appropriate Γ^{-1} diagonal values were determined

by a sequence of tests on the individual adaptive gains and sets of adaptive gains grouped according to their dynamic representation. Only the fifth adaptive gain of the thirteen values was not required for the best-case performance. The effects of other AMBC algorithm components such as Λ and the PD gains were also tested. The AMBC adaptive mechanisms compensated for these changes in either of these components and allowed for the implementation of softer PD gains with little effect on the tracking performance. The dynamic parameters of the AMBC algorithm were initialized to zero because of the lack of a priori knowledge available for the UMDH. Despite the lack of nominal dynamics, the adaptive mechanisms converged to a best-fit set of parameters that produced a best-case tracking performance. The number of training runs required for the UMDH was significantly higher than for the PUMA 560. The UMDH took 15 to 20 training runs, where as the PUMA 560 took only 4 to 6 training runs. The reason for the large disparity seems to be the complete lack of nominal dynamics used for the UMDH training and the variations in the initial state of a UMDH training run. The set of dynamic parameters which a sequence of training runs converges upon varies amongst training sequences. The tracking profiles also vary between training sequences.

Experimental evaluation of the AMBC algorithm for emulation of human finger movement produced promising results. The adaptive mechanisms of the AMBC algorithm provided compensation for the unknown UMDH system dynamics. The resulting tracking performance of the AMBC algorithm was superior to the PD controller of the baseline system. The AMBC adaptive mechanisms also proved capable of multiple trajectory tracking and provided accurate tracking whether the dynamic parameters used were from a previous trajectory or new parameters were generated for a new trajectory. The AMBC adaptation mechanism adapts to new dynamics introduced by a change in trajectory, just as it did for unknown dynamics in the original test trajectory.

Finally, the AMBC algorithm proved capable of providing payload adaptation. AMBC tracking performance with the parameters trained for no payload immediately provided tracking superior to the PD controller tracking, and further training with the payload quickly increased the disparity between the two control algorithms. Adaptation for the payload case progressed the same as for other tests in which the system dynamics were unknown. After a sequence of training runs the parameters adapted to provide improved AMBC tracking. The AMBC algorithm demonstrated an ability to compensation for unknown dynamics which allows for accurate high speed tracking, payload invariance, and multiple trajectory tracking. These are the requirements set forth for emulation of human finger movement.

5.3 Recommendations and Future Directions

Many areas of investigation may lead to improved performance of the UMDH for the emulation of human finger movement. Some of these areas are dependent on the hardware capabilities, while others rest on improved control algorithms. Probably the most obvious need for improvement is in the fingertip control of Joint 3 of the UMDH. The significant oscillations produced as a result of the control mechanisms and/or the coupling effects are a definite handicap in the performance of the robotic finger. A faster sampling rate is a likely solution to this problem. The faster sampling rate would help eliminate the over compensation of the control mechanism believed to be producing the oscillations. This would also likely improve Joint 2 performance as well. In order to increase the sampling rate, faster A/D and D/A converters must be integrated into the UMDH system electronics and the computation time of control algorithms reduced. Some of the A/D and D/A sampling time can be reduced by eliminating the retrieval of unused UMDH state variables. For the AMBC application of this thesis, the collection of the velocity and tension data could have been eliminated. However, in order to provide a complete baseline digi-

tal environment, full access to these state variables was provided for future control implementations.

Another area which may improve UMDH performance is the incorporation of some of the analog capabilities of the UMDH system into the controller. T. H. Speeter used digital control to replace the analog PD controller of the UMDH, but retained the inner most force control loop of the UMDH analog controller to help stabilize the pneumatic actuators of the UMDH, thus taking full advantage of the control unit's high speed analog response [40].

Joint drift is another problem area that should be addressed. This problem is caused by a lack of opposing tension in the joint tendons. To solve this, the cocontraction torques could be adjusted to improve drift tendencies of the robotic joints. The cocontraction torque values used in the AMBC algorithm were chosen for their optimum tracking provisions with the PD controller. The adaptation mechanisms of the AMBC algorithm, or similar tendencies of other algorithms, may eliminate the impact of the cocontraction torques on the tracking performance. This lack of dependence on the cocontraction torques for peak performance would allow the cocontraction torques to be set for optimum drift reduction.

One of the more significant difficulties encountered in the AMBC algorithm implementation was the varied initial states of a test trajectory encountered at the start of each training run. To eliminate this inconsistency, one may use the extension and flexion tension input variables to establish more regular tendon tensions before executing a trajectory. By adjusting the tendon tensions at the start of a run until the extension and flexion tension input variables are a predetermined magnitude, the initial state variations may be reduced.

Future efforts in control algorithm development for the UMDH should focus on the development and evaluation of an Artificial Neural Network (ANN) control algorithm for implementation on the UMDH. Initial research conducted on some ANN approaches theoretically suitable for the UMDH in the ARCADE_HAND en-

vironment revealed some promising possibilities. The table look-up schemes of an ANN approach such as Cerebellar Model Articulation Control (CMAC) would allow for fast response once the controller had been trained. CMAC has also been implemented on an industrial manipulator thus providing a basis for performance comparisons and a source of general application. The ADALINE controller proposed by Widrow has been implemented in simulation and also shows promise for learning the system dynamics of the UMDH.

Adoption of these recommendations will lead to control algorithms better able to meet the requirements for the emulation of human finger movement. This capability will help provide the basis for dexterous manipulation, thus taking another step towards robotic telepresence.

Bibliography

1. J.S. Albus. A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC). *J. of Dynamic Systems, Measurement, and Control*, Vol. 97:220–227, September 1975.
2. C. H. An, C. G. Atkeson, J.D. Griffiths, and J. M. Hollerbach. Experimental Evaluation of Feedforward and Computed-Torque Control. *IEEE Trans. on Robotics and Automation*, 5(3):368–372, June 1989.
3. H. Asada and J-J. E. Slotine. *Robot Analysis and Control.* John Wiley and Sons, 1986.
4. K.B. Biggers, S.C. Jacobsen, and G.E. Gerpheide. Low Level Control of the Utah/MIT Dexterous Hand. In *Proc. IEEE Conf. on Robotics and Automation*, pages 61–66, 1986.
5. R.L. Burden and J.D. Faires. *Numerical Analysis, 3rd Edition.* Prindle, Webster & Schmidt, Boston, 1985.
6. C. R. Carignan. Adaptive Tracking for Complex Systems Using Reduced-Order Models. In *Proc. of IEEE Conf. on Robotics and Automation*, pages 2078–2083, 1990.
7. L. W. Rainey III. Digital control of the utah/mit dexterous hand: Software and operation. Technical Report ARSL-90-8, Air Force Inst. of Tech., December 1990. Dept. of Elect. and Comp. Eng.
8. IRONICS Inc. IV-3201 VMEbus Multiprocessing Engine: User's Manual, 1987.
9 SARCOS Inc. Hand electronics document package, March 1987.
10. S.C. Jacobsen, E.K. Iverson, D.F. Knutti, and K. B. Biggers. Design of the Utah/MIT Dexterous Hand. In *Proc. IEEE Conf. on Robotics and Automation*, pages 1520–1531, 1986. Vol. 3.
11 S.C. Jacobsen, J.E. Woods, D.F. Knutti, K.B. Biggers, and E.K. Iversen. The Utah/MIT Dexterous Hand: Work in Progress. *The Int. J. of Robotics Research*, Vol. 4(4):21–50, Winter 1984.
12. S.C. Jacobsen, J.E. Woods, D.F. Knutti, K.B. Biggers, and E.K. Iversen. The Version I Utah/MIT Dexterous Hand. In *Proc. of the Second Int. Symp. of Robotics Research*, pages 301–308, 1984.
13. M.A. Johnson. Payload invariant control via neural networks: Development and experimental evaluation. Master's thesis, Air Force Institute of Technology, Air University, December 1989. AFIT/GE/ENG/89D-20.

14. P. K. Khosla and T. Kanade. Experimental Evaluation of Nonlinear Feedback and Feedforward Control Schemes for Manipulators. *Int. J. of Robotics Research*, 7(1):18–28, February 1988.

15. P. K. Khosla and T. Kanade. Real-Time Implementation and Evaluation of Computed-Torque Scheme. *IEEE Trans. on Robotics and Automation*, 5(2):245–253, April 1989.

16. L.G. Kraft and D.P. Campagna. A Comparison of CMAC Neural Network and Traditional Adaptive Control Systems. *IEEE Control Systems Magazine*, Vol. 10(3):884–889, April 1990.

17. T-y. Kuc and K. Nam. CMAC Based Iterative Learning Control of Robot Manipulators. In *Proc. IEEE Conf. on Decision and Control*, pages 2613–2618, 1989.

18. M. B. Leahy, Jr. Experimental Analysis of Model-Based Puma Robot Control. Technical Report ARSL-89-3, Air Force Inst. of Tech., July 1989. Dept. of Elect. and Comp. Eng.

19. M. B. Leahy Jr., D. E. Bossert, and P. V. Whalen. Robust Model-Based Control: An Experimental Case Study. In *Proc. of IEEE Conf. on Robotics and Automation*, pages 1982–1987, 1990.

20. M. B. Leahy, Jr. and K. P. Valavanis. Dynamics Based Control of Robotic Manipulators. In *Intelligent Robot Systems*. Marcel-Dekker, 1989.

21. M. B. Leahy Jr. and P. V. Whalen. Direct adaptive control for industrial manipulators. Technical Report ARSL-90-7, Air Force Inst. of Tech., October 1990. Dept. of Elect. and Comp. Eng.

22. M.B. Leahy Jr. EENG540 class notes, robotic fundamentals., Fall 1989.

23. M.B. Leahy Jr. Model-Based Auxiliary Input Control: Development and Experimental Analysis. In *Proc of IEEE CDC*, December 1990.

24. M.B. Leahy Jr. Model-Based Control of Industrial Manipulators: An Experimental Analysis. *Journal of Robotic Systems*, 7(5):741–758, October 1990.

25. R. Lippmann. An Introduction to Computing with Neural Nets. *IEEE Acoustics, Speech, and Signal Processing Magazine*, pages 4–22, April 1987.

26. M. Liu and C. D. Cook. Robust Adaptive Control for Industrial Robots - A Decentralized Systems Method. In *Proc. of IEEE Conf. on Robotics and Automation*, pages 2174–2179, 1990.

27. W.T. Miller III. Real-Time Application of Neural Networks for Sensor-Based Control of Robots with Vision. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 19(4):825–831, July/August 1988.

28. W.T. Miller III, F.H. Glanz, and L.G. Kraft. Application of a general learning algorithm to the control of robotic manipulators. *Int. J. of Robotics Research*, Vol. 6(2):84–98, Summer 1987.

29. W.T. Miller III, R.P. Hewes, F.H. Glanz, and L.G. Kranz. Real-Time Dynamic Control of an Industrial Manipulator Using a Neural-Network-Based Learning Controller. *IEEE Trans. on Robotics and Automation*, Vol. 6(1):1–9, February 1988.

30. A. Morando, R. Horowitz, and N. Sadegh. Digital Implementation of Adaptive Control Algorithms for Robot Manipulators. In *IEEE Proc. of the Int. Conf. on Robotics and Automation*, pages 1656–1662, May 1989.

31. D. Nguyen and B. Widrow. The Truck Backer-Upper: An Example of Self-Learning in Neural Networks. In *Proc. of IEEE/INNS Int. Joint Conf. on Neural Networks*, pages 357–363, 1989. Vol. 2.

32. G. Niemeyer and J-J. E. Slotine. Performance in adaptive manipulator control. In *Proc. IEEE Conf. Decision and Control*, pages 1585–1591, 1988.

33. N.Sadegh and R. Horowitz. Stability and Robustness Analysis of a Class of Adaptive Controllers for Robotic Manipulators. *Int. J. of Robot Res.*, 9(3):74–92, June 1990.

34. R. Ortega and M. W. Spong. Adaptive Motion Control of Rigid Robots: A Tutorial. In *Proc. of 27th IEEE Conf. on Decision and Control*, pages 1575–1584, 1988.

35. H.M. Schwartz, G. Warshaw, and T. Janabi. Issues in Robot Adaptive Control. In *Proc of ACC*, pages 2797–2805, May 1990.

36. H. Seraji. Decentralized adaptive control of manipulators: Theory, simulation, and experimentation. In *Proc. IEEE Conf. on Robotics and Automation*, pages 183–201, 1988.

37. H. Seraji, Lee T., and Delpech M. Experimental Study on Direct Adaptive Control of a PUMA 560 Industrial Robot. *J. of Robotic Systems*, 7(1):81–105, Feb 1990.

38. A. S. C. Sinha, S. Kayalab, and H. O. Yurtseven. Nonlinear Adaptive Control of Robot Manipulators. In *Proc. of IEEE Conf. on Robotics and Automation*, pages 2084–88, 1990.

39. J-J. E. Slotine and W. Li. On the Adaptive Control of Robot Manipulators. *Int. J. of Robotics Research*, 6(3):49–59, Fall 1987.

40. T.H. Speeter. Control of the Utah/MIT Dexterous Hand: Hardware and Software Hierarchy. *J. of Robotic Systems*, 7(5):759–790, May 1990.

41. Manx Software Systems. Manx Aztec-C, June 1988.

42. M. Tarokh. A Discrete-Time Adaptive Control Scheme for Robot Manipulators. *J. of Robotic Systems*, 7(2):145–166, Apr 1990.

43. Z. S. Tumeh. Decentralized Discrete Model Referenced Adaptive Manipulator Control. In *Proc. of IEEE Conf. on Robotics and Automation*, pages 1416–1421, 1990.

44. B. Widrow. The Original Adaptive Neural Net Broom-Balancer. In *Proc. of IEEE Int. Symp. on Circuits and Systems*, pages 351–357, 1987.

45. T. Yabuta and T. Yamada. Possibility of Neural Network Controllers for Robot Manipulators. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 1686–1691, May 1990. Vol. 3.

Vita

Captain Lloyd W. Rainey III [redacted] He graduated from Londonderry Senior High School, Londonderry, New Hampshire in 1981. He entered the United States Air Force in May 1984 under the College Senior Engineering Program. In 1985, he graduated from the University of New Hampshire with a Bachelor of Science degree in Electrical Engineering. On 13 September 1985, after attending Officer Training School, he was commissioned and and assigned to Hanscom AFB, MA as a research engineer and computer systems engineer for the Earth Sciences Division of the Air Force Geophysics Laboratory where he was responsible for the development of the Vibro-Acoustic Measurement System. In May 1989, he entered the Masters Program at the Air Force Institute of Technology, Wright-Patterson AFB, OH. He was promoted to Captain in the Regular Air Force in September 1989. In November 1988 he married Dawn Rene Dion. They now have one child, Weslie.

[redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1990	Master's Thesis

4. TITLE AND SUBTITLE

DIGITAL CONTROL OF THE UTAH/MIT DEXTEROUS HAND: INITIAL EVALUATION AND ANALYSIS

5. FUNDING NUMBERS

6. AUTHOR(S)

Lloyd W. Rainey III, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/90D-50

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AAMRL/BBA
WPAFB, OH 45433-6583

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Abstract

An experimental digital platform is developed as an environment on which to evaluate digital control strategies for dexterous manipulation with a pneumatically actuated tendon-driven manipulator. This environment is used to begin the study of advanced control methods that are suitable for providing the tracking accuracy required for grasping and dexterous manipulation with a pneumatically actuated tendon-driven end-effector. The digital platform consists of a PC/AT-386 and a single board MC68020 microcomputer in a VME chassis with shared RAM between the two processors to control the Utah/MIT Dexterous Hand (UMDH). The MC66020 controls the A/D and D/A access for the UMDH, while the ARCADE_HAND experimental control environment is hosted on the PC/AT-386 for user interface and control determinations. An Adaptive Model-Based Control (AMBC) algorithm is implemented and experimentally evaluated on the UMDH. Tracking performance is compared to the PD baseline controller of the ARCADE_HAND environment and evaluated for the requirements of human finger emulation. The evaluation includes compensation for unknown dynamics of the UMDH system, adaptability to unknown payloads, and multiple trajectory tracking capabilities. The superior tracking of the AMBC algorithm demonstrates the potential of the technique for emulation of human finger movement.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Utah/MIT Dexterous Hand, digital control, adaptive model-based control, parameter initialization, parameter convergence, dexterous manipulation, human finger emulation	78
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL